Bond Risk Premia with Machine Learning *

Daniele Bianchi† Matthias Büchner‡ Andrea Tamoni§

First draft: December 2017. This draft: April 6, 2020

Abstract

We show that machine learning methods, in particular extreme trees and neural networks (NNs), provide strong statistical evidence in favor of bond return predictability. NN forecasts based on macroeconomic and yield information translate into economic gains that are larger than those obtained using yields alone. Interestingly, the nature of unspanned factors changes along the yield curve: stock and labor market related variables are more relevant for short-term maturities, whereas output and income variables matter more for longer maturities. Finally, NN forecasts correlate with proxies for time-varying risk aversion and uncertainty, lending support to models featuring both of these channels.

Keywords: Machine Learning, Ensembled Networks, Forecasting, Bond Return Predictability, Empirical Asset Pricing

JEL codes: C38, C45, C53, E43, G12, G17.

*We are grateful to Marcus Buckmann, Hui Chen, Serdar Dinc, Winston Dou, Darrell Duffie, Chulwoo Han, Otto Van Hemert, Marcin Kacperczyk, Bryan Kelly, Zhaogang Song, Ansgar Walther, Marcin Zamojski, and participants at the 2020 AFA in San Diego, 2019 SFS Cavalcade North America at Carnegie Mellon University, the 2019 USC Dornsife INET Panel Data Conference, the 2019 Georgia State FinTech Conference, the 2019 FMA Consortium on Factor Investing in Cambridge, the conference on "New Developments in Factor Investing" at Imperial College, the workshop "Modelling with Big Data and Machine Learning" at the Bank of England, the workshop "Predicting Asset Returns" in Örebro, the 13th Imperial conference on Advances in the Analysis of Hedge Fund Strategies at Imperial College, the 29th EC^2 Conference on Big Data Econometrics, the "Alternative Risk Premia" conference at Imperial College, the 2019 FMA European Conference in Glasgow and research seminars at Durham University Business School, Lancaster University Management School and Warwick Business School for useful comments and suggestions. We thank the Centre for Scientific Computing at the University of Warwick for support with the supercomputing clusters.

†School of Economics and Finance, Queen Mary University of London, Mile End Rd, London, E1 4NS, UK. E-mail: d.bianchi@qmul.ac.uk Web: http://whitesphd.com

‡Warwick Business School, University of Warwick, Scarman Road, Coventry CV4 7AL, UK. E-mail: matthias.buechner.16@mail.wbs.ac.uk Web: http://mbuechner.com

§Department of Finance, Rutgers Business School, 1 Washington Park, Newark, NJ 07102. E-mail: andrea.tamoni.research@gmail.com Web: https://andreatamoni.meltinbit.com

1 Introduction

The recent advancements in the fields of econometrics, statistics, and computer science have spurred interest in dimensionality reduction and model selection techniques, as well as predictive models with complex features, such as sparsity and non-linearity, both in finance and economics.[1] Over the last two decades, however, the use of such methods in the financial economics literature has been mostly limited to data compression techniques, such as principal component and latent factor analysis.[2] A likely explanation for the slow adoption of advances in statistical learning is that these methods are not suitable for structural analysis and parameter inference (see Mullainathan and Spiess, 2017). Indeed, the primary focus of machine learning is prediction, i.e., to produce the best out-of-sample forecast of a quantity of interest based on a potentially large conditioning information set.

The suitability of machine learning methodologies for predictive analysis makes them particularly attractive in the context of financial asset return predictability and risk premia measurement (e.g., Gu, Kelly and Xiu, 2018). As a matter of fact, while many problems in economics rely on the identification of primitive underlying shocks and structural parameters, the quantification of time variation in expected returns is essentially a forecasting problem. This practical view complements the theory-driven approach, which often provides the building blocks for the empirical analysis of financial markets. Modeling the predictable variation in Treasury bond returns, which is the focus of this paper, provides a case in point. Forecasting excess bond returns requires a careful approximation of the a priori unknown mapping between the investors' information set and excess bond returns (e.g., Duffee, 2013, pp. 391-392).

[1] See, for example, Rapach et al. (2013), Kelly and Pruitt (2013; 2015), Freyberger, Neuhierl and Weber (2017), Giannone et al. (2017), Giglio and Xiu (2017), Heaton, Polson and Witte (2017), Kozak, Nagel and Santosh (2017), Messmer (2017), Fuster et al. (2018), Gu, Kelly and Xiu (2018), Kelly, Pruitt and Su (2018), Rossi (2018), Sirignano et al. (2018), Chen, Pelger and Zhu (2019), Feng, Giglio and Xiu (2019), Feng, Polson and Xu (2019), and Huang and Shi (2019).

[2] In economics, the initial idea of data compression techniques can probably be traced back to Burns and Mitchell (1946) who argue for a business cycle indicator that is common across macroeconomic time series. This idea was formally modeled by Geweke (1977) and Sargent and Sims (1977). Since then principal component analysis and factor analysis have been widely adopted in financial economics for forecasting problems involving many predictors(see, among others, Stock and Watson (2002a; 2002b; 2006), Forni and Reichlin (1996, 1998), Bai and Ng (2003, 2006, 2008), De Mol et al. (2008), and Boivin and Ng (2006a)).

In this paper, we employ machine learning methods to revisit the debate on the presence of predictable variation in bond returns. We work with two traditional frameworks; one that exploits information in the yield curve only, as in Cochrane and Piazzesi (2005), and one that also uses information from a dataset of hundreds of macroeconomic indicators as in Ludvigson and Ng (2009). The research design follows the structure outlined in Gu et al. (2018), whereby a comparison of different machine learning techniques is based on their out-of-sample predictive performance. Methodologically, we consider a variety of machine learning techniques to forecast excess Treasury bond returns across different maturities including partial least squares, penalized linear regressions, boosted regression trees, random forests, extremely randomized regression trees, and shallow and deep neural networks (NNs). All of these methods fall under the heading of "supervised learning" in the computer science literature.[3] Although not exhaustive, this list covers the vast majority of modern statistical learning techniques (e.g., Friedman et al., 2001). We also employ more classical dimensionality reduction techniques, such as principal component analysis (PCA), which arguably represents an almost universal approach to regression-based forecasting of Treasury bond returns.

Our contribution to the bond return predictability literature is threefold. First, within each empirical application (i.e., yields-only or yields plus macroeconomic variables), we show that non-linear machine learning methods, such as extreme trees and NNs, are useful to detect predictable variations in bond excess returns, as indicated by out-of-sample predictive R^2s that are significantly higher than those obtained by data compression techniques (e.g., linear combinations of forward rates, as in Cochrane and Piazzesi (2005), and factors extracted from macroeconomic variables, as in Ludvigson and Ng (2009)) and penalized regression techniques. Importantly, a battery of asset allocation exercises confirm that the deviations from the Expectations Hypothesis documented in this paper are economically large. In this regard, our paper contributes to the debate on the statistical evidence supporting bond return predictability (e.g., Fama and Bliss (1987), Campbell and Shiller (1991) and Cochrane and Piazzesi (2005),

[3] In "supervised" statistical learning the mapping between the quantity of interest y and the predictors $\boldsymbol{x}$ is learned by using information on the joint distribution. Unsupervised learning instead is normally implemented for data compression, e.g., PCA, and does not explicitly condition on the quantity of interest y to summarize the information content in $\boldsymbol{x}$.

for applications with yields-only) or absence thereof (e.g., Thornton and Valente, 2012).

Second, zooming in on non-linear methods, we document that using information from macroeconomic and financial variables improves the predictive accuracy of forecasts based only on (potentially non-linear transformations of) the yield curve. Indeed, the best-performing NN that exploits macroeconomic and term structure information attains out-of-sample R^2s that are about 10 percentage points larger (for maturities ranging from two to ten years) than the best-performing NN that employs yields only. Similarly, we document that employing the NN forecasts based on macroeconomic and yield information produces significantly higher certainty equivalent return values than those implied by the NN forecasts based only on yield curve information. In this respect, our paper contributes to the debate on whether there is macroeconomic variation not spanned by bond yields that helps forecast excess bond returns (e.g., see Joslin et al. (2014) for evidence in favor of unspanned macroeconomic information, and Bauer and Rudebusch (2017); Bauer and Hamilton (2018) for a critical analysis of such evidence, along with a discussion of econometric issues plaguing the "spanning" linear regressions). On one hand, our analysis reinforces the evidence in favor of unspanned macroeconomic information useful to forecast excess bond returns (e.g., Cooper and Priestley, 2009; Ludvigson and Ng, 2009; Duffee, 2011b; Joslin et al., 2014; Cieslak and Povala, 2015; Coroneo et al., 2016; Gargano et al., 2019). On the other hand, our evidence is novel in three respects. First, we continue to find support for unspanned macroeconomic risk even after accounting for potential nonlinearities in interest rates. Second, we find that it is important to account for non-linearities within macroeconomic categories in order to detect information useful for predicting excess bond returns above and beyond the yield curve. Finally, we document substantial heterogeneity in the relative importance of macroeconomic and financial variables across bond maturities: variables pertaining to the stock and labor markets are more important for short-term maturity bonds, whereas variables pertaining to orders and inventories, and output and income are more relevant for variation in long-term bonds. Thus, the type and nature of unspanned factors may depend on bond maturity.

Our third contribution concerns the economic properties of the forecasts implied by deep

NNs. First, to provide insight on the origins of the improvements in out-of-sample predictability, we investigate the ability of NNs to forecast the first three principal components of the term structure: level, slope, and curvature. We show that when using yields only as predictors, the NNs improve the forecast of the level of the term structure. However, when both macroeconomic and financial information is used in addition to yields, we find that the factors extracted from the NNs contribute to the ability to predict the level of the yield curve, as well as the slope. This is consistent with the idea that the slope of the yield curve is related to the state of the economy, and a NN is able to extract the relevant information from the large set of macroeconomic variables used. Next, we document that NN forecasts are countercyclical and mostly related to variables that proxy for macroeconomic uncertainty and time-varying risk aversion. Thus, our results support models that feature both time variation in risk prices and time-varying risk as in, e.g., Bekaert et al. (2009) and Creal and Wu (2018). However, our statistical measure of expected bond returns contrasts recent survey-based measures like the one proposed by Buraschi et al. (2019), which is mostly related to financial (specifically, bond) volatility.

In the context of machine learning in asset pricing, we document three novel facts.[4] First, our result that random forests and NNs constitute the best-performing methods even in the case when only information in the term structure is used to forecast bond returns (i.e., in a low dimensional setting) is new and provides evidence that the gain from non-linear machine learning methods is not relegated to a big data context.

Second, we show that an economically-driven choice of the network structure may perform on par with more data-driven network architectures. More specifically, when macroeconomic data are included as potential bond return predictors, we find that the out-of-sample predictive R^2 increases almost monotonically when we move from shallow specifications (one hidden layer) to deeper networks (up to three hidden layers). However, we also find that economic priors about the role of variables may improve the performance of the network. In particular, grouping variables within economic categories and then training a shallow network within each group – a

[4]The literature on machine learning and asset pricing is rapidly growing (c.f., footnote 1). Except for Huang and Shi (2019), none of these papers explore machine learning methods to forecast excess bond returns.

network structure that we dub "group ensembling"– attains a performance that is on par with the best-performing deep NN where no economic priors are utilized.[5] Thus, the depth of the network and the economic priors used to design the network (e.g., grouping within categories) interact with one another, a result that is new to the empirical finance literature.

Third, the fact that the group ensembled network outperforms more complex and agnostic specifications, is important since it highlights what type of non-linearities are important from an economic perspective: Is it the interaction of many variables (across categories) or rather a higher polynomial of the same variable (within a category)? Since our group-ensembled network switches off interactions across categories, our analysis shows that it is the non-linearity *within* a group that is ultimately relevant for the performance of the network. In this respect, our results for Treasury bond returns echo those in Gu et al. (2018) and Chen et al. (2019) for the equity market: the success of NNs lies in their ability to exploit the non-linear mapping between returns and the predictors. However, whereas Chen et al. (2019) emphasize the importance of identifying the relevant interaction between firm characteristics for equity returns, we document that, in the bond market, the interaction across economic categories matters to a lesser extent than the interaction within a category. Thus, different types of network structures may be needed for different asset markets.

The remainder of this paper is organized as follows. Section 2 provides a discussion why machine learning techniques can prove useful to measure expected bond returns within the context of predictive regression. Section 3 outlines the estimation strategy and machine learning methodologies used in the paper. Section 4 summarizes the results on predictability of bond excess returns. Section 5 dissects the predictability of bond excess returns uncovered by machine learning methods along various dimensions. Section 6 examines whether gains in predictive accuracy translate into better investment performance. Section 7 considers the economic drivers of bond return predictability. Section 8 concludes.

[5] In this respect our paper is mostly related to Huang and Shi (2019). They use an adaptive group-lasso linear regression and cluster economic variables. They show that a linear combination of such clusters significantly predicts excess bond returns. Similar to them, we explore economically motivated structures based on an ex-ante clustering. Differently from them, we focus on non-linear methodologies. In fact, our results show that linear sparse regression methods, such as lasso and elastic net, substantially underperform, statistically and economically, both shallow and deep non-linear methods, such as extreme trees and neural networks.

2 Motivating Framework

In this section, we provide a motivation for the use of machine learning to predict excess Treasury bond returns. The discussion is framed within the context of regression approaches for forecasting treasury yields. We start with the accounting identity of Campbell and Shiller (1991). We consider a zero-coupon bond with maturity $t+n$ and a payoff of one dollar. We denote its (log) price and (continuously compounded) yield at time t by $p_t^{(n)}$ and $y_t^{(n)} = -\frac{1}{n}p_t^{(n)}$, respectively. The superscript refers to the bond's remaining maturity. The (log) excess return to the n-year bond from t to $t+1$, when its remaining maturity is $n-1$, is denoted by $xr_{t+1}^{(n)} = p_{t+1}^{(n-1)} - p_t^{(n)} - y_t^{(1)}$. Then, it is possible to express the log returns to bonds as:

$$xr_{t+1}^{(n)} = -(n-1)\left(y_{t+1}^{(n-1)} - y_t^{(n)}\right) + \left(y_t^{(n)} - y_t^{(1)}\right). \tag{1}$$

The identity states that (after controlling for the slope $y_t^{(n)} - y_t^{(1)}$) any variable that forecasts the change in the bond yield from t to $t+1$, i.e., $\left(y_{t+1}^{(n-1)} - y_t^{(n)}\right)$, also forecasts the log returns to bonds. Assuming that the investors' information set at time t can be summarized by a latent k-dimensional state vector $\boldsymbol{x}_t$, and exploting the identity $y_t^{(n)} = \frac{1}{n}\sum_{j=0}^{n-1} E_t\left(y_{t+j}^{(1)} \mid \boldsymbol{x}_t\right) + \frac{1}{n}\sum_{j=0}^{n-1} E_t\left(xr_{t+j+1}^{(n-j)} \mid \boldsymbol{x}_t\right)$, we can write:

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t; N),$$

where we stack time-t yields on bonds with different maturities in a vector $\boldsymbol{y}_t$, and the maturities of the bonds in the vector N. Combining the equation above with equation (1), we obtain:

$$E_t\left[xr_{t+1}^{(n)}\right] = g(\boldsymbol{x}_t; N)\ , \tag{2}$$

for some function $g(\boldsymbol{x}_t; N)$. Every term structure model reduces to a specific mapping between yields and state variables.

In the simplest case, yields are linear affine functions of the state variables: $\boldsymbol{y}_t = A + B\boldsymbol{x}_t$.

The linearity of $f(\cdot)$, together with a dimensionality reduction of the space of yields, gives rise to principal component regression (PCR) where the quantity of interest (excess bond returns) is regressed onto principal components $\boldsymbol{x}_t$ (see Chapter 3.5 in Friedman et al., 2001):

$$E_t\left[xr_{t+1}^{(n)}\right] = \hat{\alpha} + \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_t \qquad \text{with} \qquad \boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{y}_t + b, \tag{3}$$

where the columns of $\boldsymbol{W}$ form an orthogonal basis for directions of greatest variance, and b captures the average "reconstruction error" or bias. Practically, the linear predictive system outlined in equation (3) represents a two-step procedure where researchers extract the latent factors $\boldsymbol{x}_t$, and then learn the regression coefficients $\hat{\theta} = \left(\hat{\alpha}, \hat{\boldsymbol{\beta}}^\top\right)$ by minimizing a loss function that depends on the residual sum of squares. In addition to this yields-only specification, researchers have often evaluated the role of macroeconomic variables as an important driver of bond returns. This leads to an augmented predictive regression:

$$E_t\left[xr_{t+1}^{(n)}\right] = \hat{\alpha} + \hat{\boldsymbol{\beta}}^\top \boldsymbol{x}_t + \hat{\boldsymbol{\gamma}}^\top \boldsymbol{F}_t \tag{4}$$

where $\boldsymbol{F}_t \subset f_t$ and f_t is an $r \times 1$ vector of latent common factors extracted from a $T \times N$ panel of macroeconomic data with elements $m_{it}, i = 1, \ldots, N,\ t = 1, \ldots, T$, and $r \ll N$. This is the framework originally proposed by Ludvigson and Ng (2009). Equations (3) and (4) constitute two important applications of unsupervised data compression for bond forecasting.

Another very popular set of reduced-form term structure models consists of Gaussian linear-quadratic models (e.g., Ahn, Dittmar and Gallant, 2002). In this case, the relation between yields and state variables is given by $\boldsymbol{y}_t = A + B\boldsymbol{x}_t + \boldsymbol{x}_t' C \boldsymbol{x}_t$. Note that in this case the mapping between yields and factors is non-linear in the state variables, meaning that standard linear PCA represents a mere approximation and does not necessarily give consistent estimates of the true underlying quadratic factor (see Schölkopf et al., 1998). Non-linearities are also featured in reduced-form term structure models with regime switches (e.g., Dai et al., 2007), in shadow rate models (Black, 1995; Wu and Xia, 2016), and in the model by Feldhutter et al. (2016) where the price of a bond is a time-varying weighted average of bond prices in artificial, affine

Gaussian economies.

Interestingly, non-linearity between bond yields and factors also emerges naturally in structural models with habit formation. For example, Buraschi and Jiltsov (2007) use habit formation preferences as a source of time varying market price of risk in fixed income models. In their model, yields are non-linear in the state variables, and depend on the habit stock and the factors affecting the monetary aggregate. For completeness, Appendix A provides a simple habit formation economy that leads to bond yields being a linear-quadratic function of macroeconomic variables like consumption growth, expected inflation, and habit.

Motivated by this literature, we investigate the possibility that a more precise measurement of bond risk premia can be obtained by using non-linear transformations of the data, an avenue that has also been advocated by Stock and Watson (2002a, p. 154) within the context of forecasting macroeconomic time series. Differently from the Gaussian linear-quadratic models, we do not postulate a specific functional form connecting bond yields and state variables; instead we use various statistical techniques such as trees and networks to learn about it. Besides being agnostic about the functional form between excess bond returns and macroeconomic and financial variables, the use of machine learning techniques has two additional advantages relative to the principal component regressions in equations (3) and (4).

First, the implementation of regression-based forecasts of excess bond returns using principal components as outlined in equations (3) and (4) implies that no direct use of the response variable (i.e., the excess bond returns) is made to learn about the state variables $\boldsymbol{x}_t$ and $\boldsymbol{F}_t$. This is not surprising as data compression methods such as PCA are a form of "unsupervised learning." However, equation (2) suggests that excess bond returns play the implicit role of a conditioning argument, namely, one should be able to tailor the extraction of hidden latent states $\boldsymbol{x}_t$ to the response variable $xr_{t+1}^{(n)}$. In this respect, "supervised learning" algorithms, such as the lasso, elastic net, partial least squares, regression trees, random forests, and NNs, that explicitly condition on the response variables to summarize the information in the predictors, may arguably prove useful to overcoming the limitations of standard data compression

methods.[6]

Second, traditional PCA and factor analysis (FA) are based on the assumption that all variables could bring useful information for the prediction of future excess bond returns, although the impact of some of them could be small. However, PCA or FA does not guarantee that by simply adding any number of predictors, we can be sure that the extracted factors will provide an optimal summary. Boivin and Ng (2006b) formalize this argument by providing evidence that the structure of the common components is sensitive to the input variables, and that more data does not always mean there will be more sensible estimates. In this respect, one may want to "select" the variables that actually matter for forecasting excess bond returns. Penalized regressions, such as lasso and elastic net, as well as NNs exploit the entire span of the input variables without imposing that they all carry useful information for determining excess bond returns. The existing literature on bond return predictability has vastly ignored the potential capability of machine learning techniques to address the issue of non-linearity and variable regularization. Arguably, this comes at the expense of not fully capturing the extent to which yields and macroeconomic variables are relevant for the measurement of expected excess bond returns. This is the focus of our paper.

3 Research Design

In this section, we outline the research design for the empirical analysis. We start with a description of the data, along with the specific applications. We then review the methodologies implemented in the main empirical analysis. We conclude with a short discussion of our estimation strategy.

[6]Ludvigson and Ng (2009, p. 5034) acknowledge that "factors that are pervasive for the panel of data [input] need not be important for predicting [the output]" and propose a three-step forecasting procedure where a subset of principal components extracted from a large panel of macroeconomic variables is selected according to the information criteria before running the bond return forecasting regressions. In line with this intuition, we provide evidence that supervised learning methodologies, such as NNs, are useful to exploit the information in predictors other than yields, and to improve the out-of-sample forecast of bond returns.

3.1 Data and empirical applications

The empirical analysis is based on two main benchmark applications. The first application concerns the forecasting of future bond excess returns based on the cross-section of yields as originally proposed by Cochrane and Piazzesi (2005). We use the novel zero-coupon Treasury yield curve dataset constructed by Liu and Wu (2019). This dataset allows us to study bond returns with maturity greater than five years (the longest maturity in the Fama-Bliss dataset). This is important since long maturity yields contain substantial extra predictive power over and above the first five yields (Le and Singleton, 2013, discuss the importance of using long-maturity bond yields in assessing the dynamic properties of risk premiums in Treasury markets). Moreover, Liu and Wu (2019) construct the zero-coupon curve using a non-parametric kernel-smoothing method that does not discard Treasury bills, which is instead the case for the parametric approach adopted by Gurkaynak et al. (2007). This is important since Liu and Wu (2019) find that securities at the short end of the yield curve contain important information in disciplining the overall behavior of the curve. Using the Liu and Wu (2019) yield curve dataset, we then construct forward rates and excess bond returns as described in Section 2. We focus on bonds with maturities up to 10 years. Since the U.S. Treasury started issuing 10-year notes in September 1971, this also defines the start of our sample period. We do not use bonds with longer maturities for two reasons. First, the Treasury began issuing 20-year bonds in July 1981, and 30-year bonds in November 1985. This would force us to start the analysis later, reducing further the out-of-sample period since training the NNs requires a sufficient amount of data.[7] Second, the issuance of long-maturity Treasury notes and bonds occurs at irregular intervals; to compensate for the lack of observations at long-maturities, the Liu and Wu (2019) method pulls information for the 20- and 30-year bonds from maturities that are 10 years (or more) away.

The second application consists of forecasting future bond excess returns based on both

[7] Contrary to typical machine learning applications, such as image recognition, common signal-to-noise ratios in financial data are low, exacerbating the need for sufficient data. Hence, we delay the start of the out-of-sample period so far that we have at least a handful of observations per weight to be estimated in our smallest NN (i.e., a single hidden layer with three nodes). For larger network architectures in which the number of parameters exceeds the number of available observations, regularization methods are used to ensure satisfactory training.

forward rates and a large panel of macroeconomic variables as proposed by Ludvigson and Ng (2009). We consider a balanced panel of $N = 128$ monthly macroeconomic and financial variables. A detailed description of how variables are collected and constructed is provided in McCracken and Ng (2015). The series were selected to represent broad categories of macroeconomic time series: real output and income, employment and hours, real retail, manufacturing and sales data, international trade, consumer spending, housing starts, inventories and inventory sales ratios, orders and unfilled orders, compensation and labor costs, capacity utilization measures, price indexes, interest rates and interest rate spreads, stock market indicators, and foreign exchange measures. This dataset has been widely used in the literature (e.g., Stock and Watson, 2002a, 2006; Ludvigson and Ng, 2009), and permits comparison with previous studies.

3.2 Forecasting methods

3.2.1 Principal component regressions and partial least squares

The first method we employ is a linear, dimensionality-reduction technique known as principal component regressions (PCRs). Undoubtedly, PCRs constitute the most common method used to forecast interest rates and Treasury bond returns. Unfortunately, in the classical implementation of PCRs, the target variable is discarded when extracting the latent factors. Thus, we also consider an alternative data compression methodology called partial least squares (PLS). Unlike PCR, with PLS the common components of the predictors are derived by conditioning on the joint distribution of the target variable and the regressors. Appendix E.1 provides additional details on PLS, while contrasting this method to PCRs and penalized regressions, which we discuss next.

3.2.2 Penalized regressions: ridge, lasso and elastic net

Confronted with a large set of predictors, a popular strategy is to impose sparsity in the set of regressors via a penalty term. The idea is that by selecting a subset of variables with

the highest predictive power out of a large set of predictors, and discarding the least relevant ones, one can mitigate in-sample overfitting and improve the out-of-sample performance of the linear model. In its general form, a penalized regression entails adding a penalty term on top of the OLS objective function $\mathcal{L}_{OLS}(\boldsymbol{\theta}) = \frac{1}{t}\sum_{\tau=1}^{t-1}\left(xr_{\tau+1}^{(n)} - \alpha - \boldsymbol{\beta}^{\top}\boldsymbol{y}_{\tau}\right)^2$ with $\boldsymbol{\theta} = \left(\alpha, \boldsymbol{\beta}^{\top}\right)$:

$$\mathcal{L}(\boldsymbol{\theta};\cdot) = \underbrace{\mathcal{L}_{OLS}(\boldsymbol{\theta})}_{\text{Loss Function}} + \underbrace{\phi(\boldsymbol{\beta};\cdot)}_{\text{Penalty Term}} \quad . \tag{5}$$

Depending on the functional form of the penalty term, the regression coefficients can be regularized and shrunk towards zero (as in ridge), exactly set to zero (as in lasso), or a combination of the two (as in elastic net). In Appendix E.2, we describe each method in details.

Regularized regression still does not account for non-linear relations. To address this issue, we consider a third class of non-linear methods: "shallow learners", such as regression trees and more deep structures, such as neural networks.

3.2.3 Regression trees

Regression trees are based on a partition of the input space into a set of "rectangles." Then, a simple linear model is fit to each rectangle. Figure 1 displays an example of a binary partition (Panel (a)) and the corresponding regression tree (Panel (b)). Regression trees are conceptually simple, yet powerful, and therefore highly popular in the machine learning literature. In addition to a standard regression tree methodology, we consider extensions that employ an ensemble of individual trees, like "random forests' (Breiman, 2001), and, furthermore, take into account the randomness in the predictors' partition process, like "extremely randomized trees" (Geurts et al., 2006). Appendix E.3 provides additional technical details on the estimation of regression trees (and their extensions).

3.2.4 Neural networks

Neural networks (NNs) represent a widespread class of supervised learning methods. We focus on traditional "feed-forward" networks or multi-layer perceptrons (MLP). Throughout the paper, we follow Feng et al. (2018) and adopt the convention of counting only the hidden layers, without including the output layer. For details on the estimation of NNs, see Appendix E.4. Next, we provide a high level description of the NNs structure we consider in our empirical applications.

Neural networks: yields-only. When we forecast bond returns using only information from the term structure of interest rates, we use variants of the NN architecture depicted in Figure 2. That is, we use a classical MLP. An MLP consists of, at least, three layers of nodes (the case displayed in the figure): an input layer, a hidden layer and an output layer. The result is a powerful learning method that can approximate virtually any continuous function with compact support (Kolmogorov, 1957; Diaconis and Shahshahani, 1984; Cybenko, 1989; Hornik, Stinchcombe and White, 1989). In the empirical application, we study the predictive accuracy of this classical network as we vary the number of hidden layers, as well as the number of nodes per layer.

Neural networks: macro plus yields. When we forecast bond returns using macroeconomic variables in addition to information from the term structure, we consider three alternative specifications that extend the typical MLP structure by taking into account the economic structure of the input data, as well as the nature of the forecasting problem.

The first specification, displayed in Panel (a) of Figure 3, can be thought of as a "hybrid" modeling framework in the sense that forward rates are simply included as an additional predictor in the output layer ("fwd rates direct"). This structure simulates the idea of Ludvigson and Ng (2009) in which the latent factors $\boldsymbol{F}_t$ are extracted from a large cross-section of macroeconomic variables and a linear combination of forward rates is included as proposed by Cochrane and Piazzesi (2005). We label this structure where forward rates have been pre-processed a "*hybrid* neural network".

The second specification, displayed in Panel (b) of Figure 3, *ensembles* two separate networks at the output layer level: one network is trained for the forward rates ("fwd rates net") and one for the macroeconomic variables. In contrast to the specification in Panel (a), this specification allows for a non-linear transformation of forward rates.

The third specification, displayed in Panel (c) of Figure 3, entails a collection of networks, one for each group of macroeconomic variables, which are trained in parallel and ensembled at the output layer level. The groups of macroeconomic variables are constructed following the classification provided by McCracken and Ng (2016).[8] This latter specification, which we call "*group ensembling*" switches off the (non-linear) interactions across groups of macroeconomic variables, which are instead present in the specifications of Panels (a) and (b).

To the best of our knowledge, these specifications have not been proposed before in the empirical asset pricing literature within the context of non-linear predictive methods. We study the predictive accuracy of these networks as we vary the number of hidden layers, and the number of nodes per layer.

3.3 Estimation strategy

Following common machine learning practice, we split the data into three sub-samples: a training set used to train the model, a validation set used to evaluate the estimated model on an independent data set, and a testing set, which represents the out-of-sample period in a typical forecasting exercise.[9]

We follow Gu et al. (2018) and, conditional on the estimates from the training set, we produce forecasting errors over the validation sample. We then use the prediction errors over the validation sample to iteratively search the hyperparameters that optimize the objective function. Thus, the validation sample represents the part of data that is used to provide an

[8] Specifically, we group 128 predictors into eight categories: i) output (16 series); ii) labor market (31 series); iii) housing sector (10 series); iv) orders and inventories (10 series); v) money and credit (14 series); vi) bond and FX and interest rates or financial (22 series); vii) prices or price indices (16 series); and viii) stock market (5 series). Four series in our sample could not be matched to McCracken and Ng (2016) and are left unclassified.

[9] See Chen et al. (2017) for an in-depth discussion of model fragility (i.e., the tendency of a model to over-fit the data in-sample) at the expense of poor out-of-sample performance.

unbiased evaluation of a model fit. It is trivial to see that predictions in the validation set are not out-of-sample as they are used to tune the model hyperparameters. The third sub-sample, or the testing sample, contains observations that are not used for estimation or tuning. This third sub-sample is known as the "out-of-sample" period and can be used to test the predictive performance on observations yet unseen by the machine learning model.[10]

There are a variety of splitting schemes that could be considered but the trade-off between the size of the training and validation samples is ultimately an empirical question (see Arlot et al., 2010 for a comprehensive survey of cross-validation procedures for model selection). We keep the fraction of data used for training and validation fixed at 85% and 15% of the in-sample data, respectively. The training and the validation samples are consequential. In this respect, we do not cross-validate by randomly selecting independent subsets of data to preserve the time-series dependence of both the predictors and the target variables. Forecasts are produced recursively by using an expanding window procedure, that is we re-estimate a given model at each time t and produce out-of-sample forecasts for one-year holding period excess returns.

Figure 4 provides a visual representation of the sample splitting scheme we adopt in the empirical analysis. The blue area represents the sum of the training and validation sample. The red area represents the testing sample. Notice that for some of the methodologies, validation is not required. For instance, neither standard linear regressions nor PCA require a pseudo out-of-sample period to validate the estimates. In these cases, we adopt a traditional separation between in-sample versus out-of-sample period, where the former consists of both the training data and the validation data.

We recursively forecast bond returns in excess of the short-term rate. We focus on one-year holding period excess returns for comparability with the original settings in Cochrane and Piazzesi (2005) and Ludvigson and Ng (2009).

We recursively fit machine learning methods at each time t (i.e., we increase the in-sample

[10]Note, the out-of-sample period is only "pseudo" out-of-sample in the sense that its observations are available to the researcher at the time of the study. Nevertheless, given that the observations in the out-of-sample period have not been used to train the model itself, it is standard in the machine learning literature (in asset pricing) to refer to the testing sample as the "out-of-sample" period (e.g., see Gu et al., 2018; Feng et al., 2018).

period by one monthly observation). This scheme allows us to incorporate the most recent updates from the yield curve, as well as the set of macroeconomic and financial variables.[11] When enlarging the in-sample period, we roll it forward to include the most recent information in a recursive fashion but keep constant the ratio between the training sample and the validation sample. In this respect, we always retain the entire history of the training sample, thus its window size gradually increases. By keeping the proportion of the training and validation sets fixed, the validation sample gradually increases as well. The result is a sequence of performance evaluation measures that correspond to each recursive estimate. Although computationally expensive, this leverages more information for prediction. In each empirical application, the sample spans from 1971:08 to 2018:12.

Since our forecasting exercise is iterative, the computational challenge becomes sizable. Thus, we perform all computations on a high performance computing cluster consisting of 84 nodes with 28 cores each, totaling to more than 2,300 cores. Appendix F provides a complete description of the computational specifications.

3.4 Statistical performance

We compare the forecasts obtained from each methodology to a naive prediction based on the historical mean of excess bond returns. In particular, we calculate the out-of-sample predictive R^2 as suggested by Campbell and Thompson (2007). The R^2_{oos} is akin to the in-sample R^2 and is calculated as

$$R^2_{oos} = 1 - \frac{\sum_{t_0=1}^{T-1} \left(xr_{t+1}^{(n)} - \widehat{xr}_{t+1}^{(n)} \left(\mathcal{M}_s \right) \right)^2}{\sum_{t_0=1}^{T-1} \left(xr_{t+1}^{(n)} - \overline{xr}_{t+1}^{(n)} \right)^2}, \tag{6}$$

where $\overline{xr}_{t+1}^{(n)}$ is the prediction error obtained based on the historical mean and $\widehat{xr}_{t+1}^{(n)} \left(\mathcal{M}_s \right)$ is the forecast of the excess bond returns for maturity n obtained using model $\mathcal{M}_s$, and t_0 is the

[11] Note that this is different from the implementation of machine learning methods for stock returns, where trading signals from firm characteristics are often updated once per year, which means that retraining of the models could be performed with lower frequency (Gu et al., 2018).

date of the first prediction. The first forecast error obtains by comparing the excess holding period return during the February 1989 through January 1990 period and its forecast made on January 1989.

We also build a portfolio-level return forecast from the individual maturity forecasts produced by our models. We construct the forecast of an equally-weighted portfolio by $\widehat{xr}_{t+1}^{(EW)} = \frac{1}{6}\sum_{n=2}^{10}\widehat{xr}_{t+1}^{(n)}(\mathcal{M}_s)$. We compute $R^2_{oos,EW}$ by constructing forecast errors using the realized return $xr_{t+1}^{(EW)} = \frac{1}{6}\sum_{n=2}^{10} xr_{t+1}^{(n)}$ and comparing to the historical mean.

Testing the null hypothesis, $R^2_{oos} \leq 0$, against the alternative hypothesis, $R^2_{oos} > 0$, is tantamount to testing whether the predictive model has a significantly lower mean squared prediction error (MSPE) than the historical average benchmark forecast. Thus, to test whether R^2_{oos} is significantly greater than zero, we implement the MSPE-adjusted Clark and West (2007) statistic.

A limitation of the R^2_{oos} measure is that it does not explicitly account for the risk borne by an investor over the out-of-sample period. To this, end we also calculate realized utility gains for a mean-variance or power utility investor (see Section 6).

4 An Empirical Study of U.S. Treasury Bonds

4.1 Bond return predictability and the yield curve

We start by forecasting the excess returns of Treasury bonds with the yield curve. In this case, the classical specification is given by the principal component regression (PCR) in equation (3). In words, excess returns are regressed on PCs of the Treasury term structure, i.e. $\boldsymbol{x}_t = [PC_{1,t} \ldots, PC_{k,t}]$. We use the first three, five, or ten PCs. The case with ten PCs essentially corresponds to the setting in Cochrane and Piazzesi (2005), where excess returns are regressed on a linear combination of short-rate, $y_t^{(1)}$, and nine forward rates for loans between $t+n-1$ and $t+n$, $f_t^{(n)}, n = 2, \ldots, 10$. Table 1 displays the out-of-sample R^2_{oos} (and its p-value) for different bond maturities; we also report the R^2_{oos} for an equally-weighted portfolio.

Panel A of Table 1 displays the results for PCRs and partial least squares (PLS). The first three rows show the predictive performance of PCRs for $k = 3, 5$, and 10 PCs. The predictive R^2 are negative across different maturities. A parsimonious representation with only three PCs significantly outperforms the specification with five and ten PCs, particularly at long maturities. Further, adding simple forms of non-linearities such as squared PCs worsens the performance. Perhaps surprisingly, a linear supervised learning method like PLS does not lead to any improvement relative to PCR.

Panel B of Table 1 displays the results from various configurations of the linear penalized regressions. Ridge regression performs poorly out-of-sample with predictive R^2_{oos} that are mostly negative across bond maturities. The second and third rows of Panel B show that sparse modeling improves the forecasting performance of the current term structure relative to ridge: the R^2_{oos} for both the lasso and elastic net are positive for maturities greater than four years, as well as for the equally-weighted bond portfolio. However, the performance of elastic net is on par to that obtained from a principal component regression with three PCs, which proves to be a tough benchmark.

Panel C of Table 1 shows the results for boosted regression trees, random forests, extremely randomized trees, and neural networks. All these methods attain good performance with significantly positive R^2_{oos} across maturities. With respect to trees, the randomization of the feature split locations (i.e., for extreme trees) turns out to improve the out-of-sample performance over random forests, particularly for long maturities. Turning to NNs, we observe that a shallow network with a single hidden layer and three nodes (c.f. Figure 2) performs on par with the best, deeper network with two hidden layers and seven nodes. Interestingly, further increasing the depth of the network deteriorates its performance. This continues to be the case even when we consider alternative structures, like a NN with three hidden layers and pyramidal node architecture.[12]

The last row of Panel C presents the results of an interesting case. In their paper, Cochrane

[12]In Appendix B, we employ a Diebold and Mariano (1995) pairwise test to compare the predictions from different models.

and Piazzesi (2005) conclude that lags of forward rates (dated $t-1$ and earlier) contain information about excess returns that is not spanned by month t forward rates. They note that this result is inconsistent with the logic that the time-t term structure contains all information relevant to forecasting future yields and excess returns (c.f. equation (2)). We therefore ask whether the flexibility of a NN can help reconcile the theoretical assumption that yields at time t already incorporate all information about the term structure that is needed to understand bond risk premia. To this end, the last row in Panel C reports the results obtained by feeding the NN with the ten forward rates at time t and lagged forward rates from time $t-11$ to $t-1$. By comparing the last row to the NN with one hidden layer and three nodes, we find no evidence that we can improve upon a NN that uses just the month-t forward rates.[13]

The evidence in Table 1 confirms that, even in a small dimensional setting using only information in the yield curve, we can improve bond risk premia measurement by acknowledging that (1) the function $g(\boldsymbol{y}_t; N)$ in equation (2) can be non-linear, and that (2) such improvement depends on the neural network specification; a shallow NN with one hidden layer performs on par with a network with two layers, but deeper networks worsen the performance. Finally, consistent with equation (2), we do not find extra information in the lagged values of the yield curve as opposed to just using the time-t term structure when we account for non-linearities.

4.2 Bond return predictability and macroeconomic variables

Next, we consider the set-up where information embedded in the yield curve does not necessarily subsume information contained in macro variables. In this case, the classical specification is given by equation (4), where the factors F_t now have the potential to serve as the model's state vector beyond yields only. To ensure comparability with the literature, we adopt the specification proposed by Ludvigson and Ng (2009), whereby $\boldsymbol{F}_t$ is a subset of the first eight PCs extracted from a large cross-section of macroeconomic variables and $\boldsymbol{x}_t$ represents a linear combination of forward rates as proposed by Cochrane and Piazzesi (2005), a.k.a. the

[13] We report only the best specification with lagged forward rates (i.e., a shallow NN with seven nodes). NNs with more layers or a different number of nodes underperform.

CP factor.

Panel A in Table 2 displays results from a simple principal component regression with eight PCs, the specification proposed in equation (8) of Ludvigson and Ng (2009) (i.e., $\boldsymbol{F}_t = (F_{1t}, F_{1t}^3, F_{3t}, F_{4t}, F_{8t})$), and partial least squares (PLS). Panel B shows the results from two alternative implementations of sparse and regularized linear regressions. In the first implementation ("using CP factor"), we employ the CP factor as an additional regressor; this specification ensures a closer comparability to Ludvigson and Ng (2009). In the second implementation ("using fwd rates directly"), we treat the whole set of forward rates as additional regressors with respect to macroeconomic variables. The results in Panels A and B shows that: (1) dense modeling, such as data compression techniques and ridge regression, tends to perform poorly out-of-sample; and (2) sparse modeling with both regularization and shrinkage (i.e., elastic net regressions), perform well, particularly when restricting the linear combination of forward rates. Comparing Panel B in Table 1 to that in Table 2, it seems apparent that there is information beyond the term structure of interest rates that can be used to predict bond returns.

Turning to non-linear machine learning methods, Panel C in Table 2 shows the results from the three alternative network specifications discussed in Subsection 3.2.4: (1) a "*hybrid*" framework in which the forward rates enter linearly as additional predictors in the output layer ("fwd rates direct"; Figure 3, Panel (a)); (2) a specification that ensembles one network for the forward rates ("fwd rates net") and one for the macroeconomic variables (Figure 3, Panel (b)); and (3) a specification that entails a collection of networks, one for each group of macroeconomic variables (Figure 3, Panel (c)). This latter specification, dubbed "*group ensembling*," switches off the (non-linear) interactions across groups of macroeconomic variables, which are present in the hybrid network, as well as in the specification that ensembles separately forwards and macroeconomic variables.

The performance of hybrid networks stands out. Interestingly, and differently from Table 1, increasing the depth of the NN from one- to three-layers improves its accuracy. However, a careful choice of network structure based on prior economic information exerts a great impact on

performance. In particular, a one-layer *group ensembled* model (see third-to-last row) performs on par with the three-layer *hybrid* NN for short- and medium-term maturities, and attains the highest predictive accuracy for the 7- and 10-year bonds. Interestingly, adding more layers is detrimental to the performance of the group ensembled NN.

Panel C in Table 2 also shows that the performances of boosted regression trees, random forests, and extreme trees improve substantially when using a large panel of macroeconomic information. In fact, the extreme trees performs better than shallow (hybrid and ensembled) NNs but worse than the (best performing) one-layer NN with group ensembling; see Appendix B for a formal comparison of the predictions from different models using a Diebold and Mariano (1995) pairwise test.

The results in Table 2 show that macroeconomic variables carry information that is not contained in the yield curve. When we compare the best performing (one layer and three nodes) NN in Table 1 to the 1-layer group ensembled NN, we observe approximately a 10 percentage point increase in R^2_{oos} for each maturity. The results also show that the depth and structure of the network interact with one another: having a separate network for each group of macroeconomic variables compensates for the need of a deep NN when macroeconomic variables are processed together without further classification.

5 Dissecting Predictability

5.1 Bond return predictability in expansions and recessions

We start by investigating whether bond return predictability varies over the economic cycle. To this end, we split the data into recession and expansion periods using the NBER recession indicator.

Table 3 shows the R^2_{oos} values computed separately for the recession and expansion subsamples. For yields-only PCA, we recover a classical result: predictability is concentrated in economic recessions (in particular for long maturity bonds) and is absent during expansions.

Turning to NNs, we continue to observe R^2_{oos} that are generally higher during recessions than in expansions. However, the difference in R^2_{oos} values decreases with bond maturity. More importantly, a formal test confirms that the bond return prediction from NNs is statistically different from that of the expectations hypothesis (EH) model both in expansion and recession. In contrast to NNs, the return predictability implied by trees is actually stronger during expansions. A formal test confirms that the predictive accuracy of trees is significantly better than that generated by the EH benchmark only for expansion periods. Finally, a pairwise test (untabulated) confirms that the improvement of NNs over trees is mainly due to the better predictive accuracy of networks in recessionary periods; however, the predictions from trees and NNs are indistinguishable in expansions.

Our finding that the predictability of bond returns implied by machine learning methods is not concentrated exclusively in bad times, but is present also in expansions is novel to the literature and contrasts with evidence for equities (Rapach et al., 2010; Dangl and Halling, 2012) and bonds (Gargano et al., 2019). In Appendix C.1, we analyze the models' performance in different periods using the cumulative sum of squared errors and confirm that the outperformance of machine learning-based forecasts versus the EH benchmark is not concentrated in isolated events.

Interestingly, however, it is possible to relate, ex post, NN forecasts to specific patterns of the yield curve. Appendix Table C.1 shows that NNs, and in particular the group ensemble network that exploits macroeconomic and financial information in addition to interest rates, predict high excess bond returns when there is a steep slope in the yield curve (e.g., right after recessions) and when the level of the yield curve is high. Further, these NN predictions (conditional on specific shapes of the yield curve) are highly correlated with realized returns, thus leading to high R^2s.

5.2 Understanding the performance of neural networks: level, slope or both?

In this subsection we provide an heuristic interpretation of the performance of the NNs based on the Campbell and Shiller (1991) accounting identity (equation (1)). Such identity posits that the forecasts of future yields (or their PCs) using current yields are necessarily also forecasts of expected log returns to bonds. Thus, we investigate the ability of the latent factors extracted by the NN to predict the year-on-year changes in the first three PCs extracted from the cross section of forward rates. We denote these principal components as $PC_{1,t}, PC_{2,t}$, and $PC_{3,t}$. The auxiliary forecasting regressions are:

$$PC_{i,t+1} - PC_{i,t} = b_0 + \boldsymbol{b}_1^\top \mathcal{P}_t + \boldsymbol{b}_2^\top \boldsymbol{x}_t + \epsilon_{i,t+1} \qquad \text{for} \quad i = 1, 2, 3,$$

where we stack the first three PCs of the term structure in the vector $\mathcal{P}_t$ and denote by $\boldsymbol{x}_t$ the hidden factors extracted by the NN.[14]

Table 4 reports the in-sample R^2 of such predictive regressions. The first row provides the benchmark results based on the sole vector $\mathcal{P}_t$. In support of Duffee (2011a, 2013), we find weak evidence that changes in the first PC (level) are forecastable (R^2 = 9.28% being the lowest), whereas the slope and curvature are unquestionably forecastable with 21.66% (48.70%) of the variation in slope (curvature) that is predictable.

Next, we add to the regression the hidden factors extracted from the two best performing NNs in Table 1-2: the *NN 1 Layer (3 nodes)* – when forecasting only with the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables. We observe that the factors extracted from NNs that use yields-only (second row in Table 4) contribute substantially to the predictability of the level and curvature. On the other hand, the statistical evidence for slope forecasts - after controlling for the standard three principal

[14] Take a shallow network with $L = 1$ hidden layers as an example, i.e.,: $E_t\left[xr_{t+1}^{(n)}\right] = \hat{\alpha}_n + \hat{\boldsymbol{\beta}}_n^\top \boldsymbol{x}_t$, where, $\boldsymbol{x}_t = h\left(\boldsymbol{W}\boldsymbol{y}_t + b\right)$. The latent factor $\boldsymbol{x}_t$ is extracted at each time t conditional upon estimates of the weights $\boldsymbol{W}$ and bias b from the vector of inputs y_t.

components - is weak. We conclude that there is substantial information in the time-t term structure not only about future values of slope, but also about the level (and curvature). Standard PCs are not entirely able to extract all the relevant information about the level. A shallow NN is successful in extracting such information about the future level of the curve, an information which leads to excess returns being more predictable out-of-sample.

The last row in Table 4 focuses on factors extracted from a NN that exploits macroeconomic variables in addition to forward rates. We observe that the factors extracted from the group ensembled NNs not only contribute to the ability to predict the level of the yield curve, but also the slope. This suggests that the slope of the yield curve is related to the state of the economy, and our NN is able to extract the relevant information from our large set of macroeconomic variables.

5.3 Relative importance of macroeconomic variables

In this subsection, we investigate which variables drive the performance of NNs studied in Tables 1-2. To this end, we examine the marginal relevance of single variables based on the partial derivative of the target variable, $xr_{t+1}^{(n)}$, with respect to each input, where the gradient is evaluated at the in-sample mean value of the input, i.e.,

$$\mathbb{E}\left[\frac{\partial}{\partial y_{it}} xr_{t+1}^{(n)} \middle| y_{it} = \overline{y}_i\right], \tag{7}$$

where $\overline{y}_i$ represents the in-sample mean of the input variable i. The partial derivative represents the sensitivity of the output to the ith input evaluated at its sample mean, conditional on the network structure and the average value of the other input variables (see Dimopoulos et al., 1995). Further, we focus on the magnitude of the gradients by taking their absolute values. Appendix C.3 provides additional discussion about the computational details.

Figure 5 shows the relative importance of each input variable based on the gradient in equation (7). The analysis is carried out for the best-performing NN in Table 2, the NN with one hidden layer where macroeconomic variables and forward rates are modeled separately

through ensembling at the output layer level. For ease of exposition, we report only the top 20 most relevant predictors. Panels (a) and (b) display results for the 2- and 10-year bond maturities, respectively. To gain further intuition about the systematic patterns in the drivers of expected excess bond returns, we also calculate the relative importance from the absolute gradients averaged for each class of input variables as labeled in McCracken and Ng (2016). The results provide an indication of which economic category dominates.

In Panel (c) and (d) of Figure 5, the variables pertaining to inflation, and money and credit are important independently from the maturity considered. However, the results also show that the effect of other classes of predictors is heterogeneous over the term structure. For instance, variables related to the stock and labor market are more important for the short-end of the yield curve, while variables pertaining to the categories output & income and orders & inventories become more relevant for the long-end of the curve. This analysis is important for two reasons. First, it suggests that inflation has a level-like effect on bond yields, whereas variables pertaining to the labor market (order & inventories) are likely to have a slope effect acting mostly on short-term (long-term) bonds, while leaving long-term (short-term) bonds unaffected. This evidence can therefore provide guidance for theoretical models that include macro risk factors as drivers of bond risk premia by highligting their permanent or transient nature. Second, our analysis suggests that the use of excess bond returns averaged across maturities is unlikely to flesh out the true impact of macro risk factors on bond risk premia.[15]

In all, our results show that there is information in macroeconomic and financial variables beyond that conveyed by the yield curve, and this information improves the predictions of bond returns (Tables 1-2). In addition, the type of unspanned (by the yield curve) information may vary across different bond maturities. To our knowledge, this fact is novel and provides a new angle to revisit a central question in the term structure literature, which is whether yields data contain all the relevant information to predict future bond returns.

[15]Our evidence of a level-like effect of inflation on bond yields is in line with the analysis in Joslin et al. (2014, Section VI). Furthermore, within the orders & inventories category, we find that "New Orders for Durable Goods" is of single-most importance for forecasts at the far end of the curve (see Panel (b) in Figure 5). Yang (2011) provides theoretical and empirical evidence that is consistent with our finding by showing that the impact of durable consumption growth on the yield curve strengthens with bond maturity.

5.4 Interactions within or across categories?

The finding that a shallow group ensembled network performs on par with (or better than) a deep, three layer NN that models all macroeconomic and financial variables together is novel to the literature on machine learning and asset prices, and is important for two reasons. First, our results on group ensembled NNs show that the depth of the network and the economic priors used to design it (e.g., grouping variables that pertain to the same category) interact with one another. In particular, for the application at hand, group ensembling can compensate for the depth of the network. Second, our results highlight what type of non-linearities are important from an economic perspective: Is it the interaction of many variables (across categories) or a higher polynomial of the same variable (within a category)? Since our group ensembled network switches off interactions across categories, our results show that it is the non-linearity within a group that drives the outperformance of the network. Of course, this is true only in so far as cross-group network weights are not already small in the fully connected network. Table C.2 in the Appendix shows that this is indeed not the case. More precisely, we calculate the second order derivatives of the output with respect to each input conditional on the inputs being in different groups of predictors.[16] We estimate cross-group partial derivatives for both a fully connected network and a network with group-ensembling. The results in Panel A show that the absolute value of the sum of interaction derivatives in the fully connected network is orders of magnitude larger than the value obtained from a group-ensembled NN (i.e., the cross-group interactions are indeed large in the fully connected network). In Panel B of Table C.2, we provide the within-group second order partial derivatives of the outputs with respect to the inputs conditional on being in the same group. We find that the magnitude of the within-group effects is similar between the fully connected and group-ensembled NNs. Hence, the performance of the group-ensembled NN is driven by imposing the absence of interactions across categories while allowing for non-linearity within an economic category.

[16] That is, we calculate:

$$\mathbb{E}\left[\frac{\partial^2}{\partial y_i \partial y_j} xr_{t+1}^{(n)} \middle| y_i \in G_A, y_j \in G_B\right], \tag{8}$$

where G_A and G_B are two non-overlapping groups of variables defined as in McCracken and Ng (2016), and we sum the absolute value of equation (8) for each interaction of variables that do not belong to the same group.

5.5 Model uncertainty

Faced with multiple NN estimates, the question of how to best exploit ex-ante different forecasting specifications immediately arises. In particular, should we rely on a single, ex post, dominant model specification or should a combination of different forecasts be used to produce a better forecast? From a pure theoretical perspective, unless the best forecasting model can be identified ex-ante, forecast combinations may offer some diversification benefits (see Clemen, 1989, for a discussion). However, it may also be the case that a carefully designed validation procedure is able to systematically pick the best out-of-sample model specification.

To answer this question, we first compare the best-performing NN within the context of forecasting bond returns with both yields and macroeconomic variables – the *NN 1 Layer Group Ensem + fwd rate net* (see Table 2) – against a combined forecast of the form,

$$\hat{xr}_{c,t+1}^{(n)} = \sum_{i=1}^{\mathcal{M}} \omega_{i,t} \cdot \hat{xr}_{i,t+1}^{(n)} \tag{9}$$

where $\hat{xr}_{c,t+1}^{(n)}$ denotes the one-step ahead combined forecast for maturity n, $\omega_{i,t}$ is the weight assigned to each individual prediction, $\hat{xr}_{i,t+1}^{(n)}$, and $i = 1, \ldots, \mathcal{M}$ are the forecasts from the set of NNs $\mathcal{M}$ in Table 2. We choose two representative model combination schemes: (1) an equal weight assigned to each forecast, i.e., $\omega_{i,t} = 1/\mathcal{M}$, and (2) a linear combination of forecasts based on the validation losses, i.e., $\omega_{i,t} = \frac{1/L(e_{i,t}|\theta_i)}{\sum_{i=1}^{\mathcal{M}}(1/L(e_{i,t}|\theta_i))}$, where $L\left(e_{i,t}|\theta_i\right)$ is the validation loss obtained from the cross-validation prediction error $e_{i,t}$ given the network hyperparameters θ_i.[17]

In addition to an equal-weight and a relative-performance combination scheme we also compare our best-performing NN forecasts against a full-blown cross-validated network. In particular, we expand the set of hyperparameters that are cross-validated and selected every five years; that is, we let optimization procedures not only select the dropout rate and the L1/L2 penalties, but also the number of hidden layers, the nodes per group of macroeco-

[17]Note that the loss function we use is a simple mean squared error plus a penalty to induce regularization in the weights. This means that the weighting scheme reflects the performance of each model relative to the performance of the average model (e.g., Bates and Granger, 1969; Newbold and Granger, 1974; Stock and Watson, 1998; Elliott and Timmermann, 2004).

nomic variables, and the nodes in the forward rate network (see Table F.1 for details on the hyperparameters).[18]

The logic for comparing our best-performing model against two representative forecast combination schemes and a full-blown cross-validated network is to make sure that our results are robust to more flexible and adaptive modeling strategies. Table C.3 in Appendix C.5 reports the results. Two interesting aspects emerge from the table. First, the group-ensembled NN (see first row) outperforms both forecast combination schemes (second and third rows), the sole exception being at the two-year maturity. Second, our group-ensemble network specification tends also to perform on par with, or better than, the full-blown cross-validated network for maturities greater than three years. Hence, we conclude that the optimal structure of layers and nodes, which is endogenously chosen through an adaptive cross-validation exercise, does not improve (except for the very short-end) upon a more parsimonious and economically motivated network structure, like our one-layer group-ensemble NN.

We next examine the recursive performance, meaning cross-validation error, of the top performing neural network. In principle, the performance of the *NN 1 Layer Group Ensem + fwd rate net* specification could be justified by the fact that such network is consistently chosen through cross-validation and across time. In Table C.4 in Appendix C.5, we compare how often the four on average best-performing NN structures from Table 2 are selected throughout the out-of-sample period. Two interesting facts emerge. First, our best-performing group-ensembled NN generates the smallest validation error (and thus it would have been chosen through cross-validation) for about a half of the out-of-sample period. This could explain the similar performance between the best-performing (group-ensemble) NN in Table 2 and the full-blown cross-validate model, which contains the benchmarking specification in the model set. Second, shallow NNs tend to consistently deliver lower validation errors. This reinforces our result that network depth and structure interact with one another: in fact, a carefully designed network outperforms a deeper, and more data-driven, network structure.

[18] We thank an anonymous referee for suggesting this exercise.

6 Economic Value of Excess Bond Return Forecasts

So far, our analysis concentrated on statistical measures of predictive accuracy. Next we evaluate whether the apparent gains in predictive accuracy translate into better investment performance relative to the no-predictability alternative. This is important since Thornton and Valente (2012) find that yield-based predictors, when used to guide the investment decisions of an investor with mean-variance preferences, do not lead to higher out-of-sample Sharpe ratios compared with investments based on a no-predictability expectations hypothesis (EH) model. Sarno et al. (2016) reach a similar conclusion. However, the large amount of time variation in expected bond returns that is detectable in real time by machine learning methods naturally calls for revisiting these findings.

6.1 The asset allocation framework

In order to assess the economic importance of machine learning methods (particularly trees and NNs) in forecasting bond returns, we use a classic portfolio choice problem (Della Corte et al., 2008; Thornton and Valente, 2012). Specifically, we consider an investor who optimally invests in a portfolio comprising $K+1$ bonds: a risk-free one-period bond and K risky n-period bonds.

We consider both univariate and multivariate asset allocation exercises. In the univariate case, the investor selects between an n-year bond and the risk-free return based on the expected return implied by a given model. We focus on the results for $n = 2$ and $n = 10$ years. In the joint asset allocation exercise, the investor selects bonds with maturities of two- to ten-years, and the risk-free return. We analyze the asset allocation decisions of a mean-variance investor and those of a power utility investor. The discussion of the power utility problem and its solution is in Appendix D.1, and in the remaining discussion focus on the mean-variance case.

At each time t, the decision-maker selects the weights on the risky n-period bonds $\boldsymbol{w}_t = \left[w_t^{(2)} \dots w_t^{(10)}\right]$ to maximize the quadratic utility:

$$\max_{\boldsymbol{w}_t} \; E\left[R_{p,t+1}\right] - \frac{\gamma}{2}\text{Var}\left(R_{p,t+1}\right),$$

where γ is the risk aversion coefficient of the mean-variance investor, $R_{p,t+1} = 1 + y_t^{(1)} + \boldsymbol{w}_t' \boldsymbol{xr}_{t+1}$ is the gross return on the portfolio, $E\left[R_{p,t+1}\right]$ is the sample mean portfolio return, and $\text{Var}\left(R_{p,t+1}\right)$ is the sample variance portfolio return. Then the solution of the above optimization is $\boldsymbol{w}_{t,s} = \frac{1}{\gamma}\Sigma_{t+1|t}^{-1}\widehat{\boldsymbol{xr}}_{t+1}\left(\mathcal{M}_s\right)$, where $\widehat{\boldsymbol{xr}}_{t+1}\left(\mathcal{M}_s\right)$ is the vector of bond returns' forecast obtained using model $\mathcal{M}_s$, and $\Sigma_{t+1|t} = Var_t\left(\boldsymbol{xr}_{t+1} - E_t\left[\boldsymbol{xr}_{t+1}\right]\right)$. For the univariate allocation exercise we have: $w_{t,s}^{(n)} = \frac{\widehat{xr}_{t+1}^{(n)}(\mathcal{M}_s)}{\gamma\sigma_{t+1|t}^{(n)}}$ where $\widehat{xr}_{t+1}^{(n)}\left(\mathcal{M}_s\right)$ is the bond returns' forecast for maturity n given model $\mathcal{M}_s$, and $\sigma_{t+1|t}^{(n)}$ the diagonal element of $\Sigma_{t+1|t}$ relative to the bond with n-year maturity.

To proxy for $\Sigma_{t+1|t}$, we employ a rolling sample variance estimator as in Thornton and Valente (2012): $\widehat{\Sigma}_{t+1|t} = \sum_{l=0}^{\infty}\Omega_{t-l} \odot \epsilon_{t-l}\epsilon_{t-l}'$, where $\epsilon_t = \left[\epsilon_t^{(2)} \ldots \epsilon_t^{(10)}\right]'$ are forecast errors, $\Omega_{t-l} = \alpha\exp\left(-\alpha\right)\mathbf{11}'$ is a symmetric matrix of weights, $\odot$ denotes element-by-element multiplication, and we set the decay rate α to 0.05 (same value as in Thornton and Valente (2012) and within the range of those reported in studies like Fleming et al. (2001)). We also winsorize the weights for each of the n-period bonds to $-1 \leq w_t^{(n)} \leq 2$ to prevent extreme investments; however, we evaluate the robustness of our results to alternative assumptions about the portfolio weights. Finally, to make our results directly comparable to other studies (e.g., Thornton and Valente, 2012; Gargano et al., 2019), we assume a coefficient of risk aversion of five.

Given the Markowitz optimal weights on the risky bonds, we compute the realized utilities. Then, following Fleming et al. (2001), we obtain the certainty equivalent gains (annualized and in percentages) by equating the average utility of the EH model with the average utility of any of the alternative models.

To test whether the certainty equivalent return (CER) values are statistically greater than zero, we use a Diebold and Mariano (1995) test. Specifically, to evaluate the allocation implied by the NN forecasts, we estimate the following regression:

$$u_{t+1,NN} - u_{t+1,EH} = \alpha^{(n)} + \varepsilon_{t+1}\ ,$$

where $u_{t+1,s} = \boldsymbol{w}_{t,s}'\boldsymbol{xr}_{t+1} - \frac{\gamma}{2}\boldsymbol{w}_{t,s}'\Sigma_{t+1}\boldsymbol{w}_{t,s}$ and $s = \{EH, NN\}$, i.e., we use the optimal weights together with the realized returns.

6.2 Asset allocation: results

Table 5 shows the annualized CER values computed relative to the EH model. Positive values indicate that the predictive model performs better than the EH model. We focus on the best predictive models from Tables 1-2, the extreme trees and the *NN 1 Layer (3 nodes)* – when forecasting only with forward rates – or *NN 1 Layer Group Ensem + fwd rate net* – when including the macroeconomic variables.

With the sole exception of the mean-variance investor selecting the two-year bond, the remaining CER values for the trees and NNs are significantly higher than those generated by the EH benchmark. The CER values increase with bond maturity, but the highest CER values are found in the multivariate setting, which suggests that the economic gains associated with NNs forecasts are not limited to specific maturities.

Interestingly, when the (mean-variance or power utility) investor makes no use of information beyond the term structure of interest rates, then trees deliver CER values that are 0.2%-0.7% greater than those obtained using NNs. However, when the investor also considers information from macro and financial variables, then NNs outperform trees by 0.5% (power utility and 10-year bond) to 1% (multivariate setting). A pairwise test confirms that this improvement of NNs over trees is statistically significant. Furthermore, the results in Table 5 also show that (for the univariate and multivariate allocation, independently from utility) the group-ensemble NN that exploits macroeconomic information produces significantly higher CER values than those implied by the best-performing NN using yields-only. Overall, our machine learning-based forecasts of bond returns provide support for the hypothesis that a (statistically and economically) significant portion of macroeconomic information is not captured by the yield curve, even after accounting for non-linearity in interest rates.[19]

Appendix Table D.1 shows the investment performance when we change assumptions about the portfolio weights. In the first scenario (Panel A), we restrict the weights on the risky bonds

[19]The p-values based on power utility (Panel B) are lower than those reported for mean-variance (Panel A). This is because the power utility setting generates less volatile CER series. To address the higher persistence of power utility CER, we compute Newey-West standard errors using a larger truncation parameter equal to 20 lags (see Lazarus et al., 2018). Even in this case, we continue to find statistical support for our conclusions.

to the interval $[0,\ 0.99]$ to ensure that the expected utility is finite even with an unbounded return distribution (e.g., Kandel and Stambaugh, 1996; Geweke, 2001a). In the second scenario (Panel B), we leave the portfolio weights unrestricted and instead restrict the bond returns to fall between −100% and 100% to prevent the expected utility from becoming unbounded (e.g., Johannes et al., 2014). In both cases, our conclusions continue to hold: the CER values of the tree and NN models are generally significantly higher than those generated by the EH benchmark; moreover, NNs outperform trees provided that macroeconomic variables are included in the set of predictors.

In summary, we find that a NN that exploits the non-linearities within groups of macroeconomic variables delivers high predictive accuracy (see Table 2), which, in turn, translates into investment strategies with large economic value (see Table 5).

7 Economic Drivers of Bond Return Predictability and Portfolio Performance

In this section, we investigate whether our forecasts of excess bond returns are consistent with explanations based on time-varying risk premia. We then examine the economic drivers of bond return predictability and portfolio performance.

7.1 Cyclical pattern of expected excess bond returns

We start by investigating the cyclicality of our forecasts of excess bond returns. Indeed, standard asset pricing models featuring habit persistence like Wachter (2006) suggest that bond risk premia are countercyclical.

In Panels (a) and (b) in Figure 6, we plot the forecast of 10-year bond returns obtained from the best-performing NNs against the industrial production index growth. The results are similar for alternative maturities. We report the prediction based on yields only (Panel (a)),

as well as the prediction obtained by adding macroeconomic variables to forward rates (Panel (b)). In Panels (c) and (d), we overlay our forecasts with the realized ten-year excess bond returns series.[20]

Independently of the set of predictors employed, Panels (a) and (b) in Figure 6 reveal that the bond risk premium obtained from NNs displays a clear countercyclical pattern. In particular, the contemporaneous correlations between forecasts of the ten-year excess bond returns and industrial production is −12.4% (p-value of 0.07) when only information in the term structure is used (Panel (a)). This correlation almost doubles to −24.6% (p-value of 0.01) when we add macroeconomic variables to forward rates (Panel (b)).[21] Thus, using macroeconomic variables greatly improves the estimates of the risk premium.

This prima facie evidence suggests that our forecasts may be consistent with the fact that investors want to be compensated for bearing recession-related risks. To the extent that our forecasts of excess bond returns reflect time-varying risk premia, we would also expect higher Sharpe ratios in recessions. To this end, Table 6 reports, for the 2- and 10-year bond maturities, the Sharpe ratios computed separately for recession and expansion periods. We find that, across all maturities and forecasting models, the Sharpe ratios are substantially higher during recessions than in expansions.

7.2 Economic drivers of expected excess bond returns

Having established that our forecasts of excess bond returns, and the associated Sharpe ratios, move counter-cyclically, we next investigate whether these forecasts are linked to key drivers of bond risk premia suggested by asset pricing theory and previous evidence. In particular, we regress the forecasts of ten-year excess bond returns obtained from the best-performing NNs in Tables 1-2 on a set of structural risk factors that arise in equilibrium models and generate time-varying bond risk premia. Each row in Table 7 corresponds to a different speci-

[20] Relative to yields-only, the addition of macroeconomic variables leads to: (1) NN forecasts that are higher in the recession of 2007-2009, and (2) better predictive performance. In Appendix C.1, and in particular Panels (b) and (d) of Figure C.1, we examine the predictive accuracy of NNs throughout our sample period.

[21] The correlation p-values are computed using Newey and West (1987) standard errors with 12 lags.

fication.[22]

Motivated by the literature on the role of disagreement in asset prices (e.g., Buraschi and Jiltsov, 2007; Dumas et al., 2009), we examine the role played by differences in beliefs for the dynamics of excess bond returns. The results are in row (i) in Table 7. We proxy for real disagreement (DiB(g)) and nominal disagreement (DiB(π)) using the interquartile range of four-quarter-ahead forecasts of GDP and consumer prices (CPI), respectively, obtained from the Survey of Professional Forecasters (SPF).

We investigate the link between time-varying risk aversion and excess bond returns in rows (ii) and (iii) of Table 7. Asset pricing models featuring habit persistence suggest that risk premia should be higher during recessions due to a reduced surplus consumption ratio. Following Wachter (2006), we proxy for risk aversion using (the negative of) a weighted average of 10 years of quarterly consumption growth rates (dubbed –Surplus). We also employ the new measure of time-varying risk aversion proposed by Bekaert et al. (2019) (dubbed RAbex). This risk aversion measure is calculated from observable financial information at high frequencies.

We next examine the role played by economic growth and inflation uncertainty, UnC(g) and UnC(π), for expected bond returns in row (iv). This link can be motivated by long-run risk models like Bansal and Shaliastovich (2013) or by habit-models that allow for time variation in quantities of risk like Creal and Wu (2018).[23]

Finally, we examine the link between bond volatility and our forecasts of excess bond returns in row (v) of Table 7. To assess this link, we employ two proxies: (1) the intra-month sum of squared yield changes (returns) on a constant maturity 10-year zero-coupon bond (denoted as $\sigma(n)$); and (2) the one month implied 10-year maturity bond risk-neutral volatility published by the CME (denoted as $TYVIX$).[24]

[22] We focus on the 10-year maturity bond to make a clean comparison with Buraschi et al. (2019). However, we find our conclusions to be robust across maturities.

[23] To proxy for uncertainty, we adapt the procedure of Bansal and Shaliastovich (2013). In the first step, we use our SPF on consensus expectation of four-quarter GDP growth and inflation and fit a bivariate VAR(1). In a second step, we compute a GARCH(1,1) process on the VAR residuals to estimate the conditional variance of expected real growth and inflation.

[24] https://www.cboe.com/products/vix-index-volatility/volatility-on-interest-rates/cboe-cbot-10-year-u-s-treasury-note-volatility-index-tyvix.

Several conclusions emerge from the results in Table 7. First, the link between structural risk factors and realized returns is generally weak. The sole factor that is statistically linked to realized bond returns is the risk neutral volatility (Panel A, row v).

Our forecasts of excess bond returns paint a completely different picture. Independently from the set of predictors we use, we find a strongly positive coefficient on uncertainty about economic growth but not on inflation uncertainty (Panels B and C, row iv). We also find strong support for the prediction of equilibrium models based on habit preferences (Panels B and C, row ii). Adding macroeconomic information strengthens this conclusion: in this case, the slope coefficient on the risk aversion measure proposed by Bekaert et al. (2019) is also positive and statistically significant (Panel C, row iii). Finally, in line with Duffee (2002), we find only a weak link between expected bond returns and bond volatility (Panels B and C, row v; the link is marginally significant in Panel B but the R^2 is small).

There are minor differences between Panel B and C. In particular, the addition of macroeconomic variables leads to a positive and statistically significant slope coefficient on nominal disagreement (Panel C, row i). However, in a horse race only (habit) risk aversion and macroeconomic uncertainty continue to stay significant, leading to a large R^2 of about 25%. Instead, (nominal) disagreement is driven out (Panel C, row vi). This is also the case in Panel B. Comparing row (vi) in Panels B and C to Panel A, it is apparent that using the measure of excess bond returns implied by NNs instead of realized returns leads to stronger support of the predictions of equilibrium models.[25]

In Appendix D.3, we discuss the relation between the realized utility obtained in the portfolio analysis in Section 6 and the same structural risk factors presented in this section. The results in Tables D.2-D.3 show that the relation between utility gains from our portfolio analysis is the strongest with risk aversion and time-varying uncertainty.

The evidence in Table 7 for bond returns forecasts, and that in Tables D.2-D.3 for realized utility, confirm that the variation in expected bond returns implied by a NNs can be understood

[25] A saturated regression that includes all variables simultaneously leads to the same conclusion: only habit-based risk aversion and macroeconomic uncertainty remain significant. The R^2 from the saturated regressions are 12%, 25.51%, and 25.60%, adding little explanatory power to the specification (vi) in Panels A, B, and C.

in terms of time variation in risk prices and time-varying (macroeconomic) risk. Overall, our results support models that feature both channels, such as Bekaert et al. (2009) and Creal and Wu (2018).

Our evidence stands in stark contrast to the recent finding of Buraschi et al. (2019). They find that the quantity of risk as measured by bond volatility has a strong role, whereas habit-based risk aversion matters little. Thus, our statistical measure of bond risk premia likely captures a potentially different channel component from the subjective bond risk premia of Buraschi et al. (2019). We investigate this point further in the next subsection.

7.3 Statistical versus subjective forecasts

Table 8 reports the correlations between our forecasts of ten-year excess bond returns obtained from the best-performing tree and NNs in Tables 1-2 with three recent proxies for risk premia that rely on interest rates forecasts as surveyed by the Blue Chip Financial Forecasts (BCFF): (1) the measure of subjective bond risk premia (EBR^*) proposed by Buraschi et al. (2019) based on aggregation of expectations of (the top decile of) professional forecasters; (2) the Piazzesi et al. (2015) consensus measure of subjective bond risk premia constructed as the difference between subjective and VAR interest-rate expectations, $E_t^*\left[i_{t+h}^{(n-h)}\right] - E_t\left[i_{t+h}^{(n-h)}\right]$; and (3) the forecasts by Giacoletti et al. (2016) based on a learning rule that updates beliefs using the history of bond yields and disagreement among forecasters.[26,27]

The results in Table 8 show that, across all bond maturities and model specifications, the correlation between our forecasts and the subjective bond risk premia of Buraschi et al. (2019) is small, and not statistically significant. This is in line with our previous analysis of economic drivers of bond return predictability (Table 7): our forecasts are associated with proxies for

[26] We kindly thank Paul Whelan and Marco Giacoletti for sharing their proxy of bond risk premia with us.

[27] We measure $E_t^*\left[i_{t+h}^{(n-h)}\right]$ using the median survey forecast of $i_{t+h}^{(n-h)}$ for the 10-year Treasury bond from the Survey of Professional Forecasters. Importantly, Piazzesi et al. (2015) find that "median forecasts from the SPF are similar to those from the Bluechip survey." The statistical forecasts follow Piazzesi et al. (2015): we compute the forecasts by running OLS directly on the system $Y_{t+h} = \mu + \phi Y_t + \varepsilon_{t+h}$, so that we can compute the h-horizon forecast simply as $\mu + \phi Y_t$. The vector of interest rates Y includes the 1-, 2-, 3-, 4-, 5-, 7-, and 10-year maturity.

time-varying risk aversion and macroeconomic uncertainty whereas Buraschi et al. (2019) find a strong link between the quantity of risk channel (as proxied by bond volatility) and their proxy for bond risk premia.

We instead find a strong and positive association between our forecasts and the Piazzesi et al. (2015) measure of bond risk premia, in particular when yields-only based forecasts are considered. This is perhaps not surprising given the evidence in Buraschi et al. (2019): the consensus is not a sufficient statistic for the cross-section of expectations so that aggregation of subjective bond risk premia for each single contributor (as in $EBR^{\star}$) may differ from measures that rely on the simple arithmetic average of the cross-section of forecasters (as in Piazzesi et al., 2015). However, we note that adding macroeconomic variables weakens the correlation between our forecasts and subjective measures based on consensus.

Finally, the correlation between our forecasts and those of Giacoletti et al. (2016) are quite large and mostly significant. This effect is generally stronger for long maturity bonds and when yields-only based forecasts are considered. Overall, dynamic learning effects could account for some of our findings of bond return predictability.

8 Conclusion

In this paper we evaluate the benefits of using machine learning methods for understanding bond price fluctuations. Three main findings emerge from our analysis.

First, we show that non-linear machine learning techniques, such as extreme trees and neural networks, detect predictable variations in bond returns that are statistically large; importantly, the forecasts implied by these methods translate into similarly large out-of-sample economic gains. Second, we document that employing the NN forecasts based on macroeconomic and yield information produces significantly higher certainty equivalent return values than those implied by the NN forecasts based on yields-only variables, thus providing support for information that is unspanned by (potentially non-linear transformations of) the yield curve and

yet useful to forecast bond returns. We also provide evidence of a significant heterogeneity in the relative importance of macroeconomic variables across bond maturities. Hence, the type and nature of unspanned factors may depend on the bond maturity. Finally, we document that NN forecasts are countercyclical and mostly related to variables that proxy for macroeconomic uncertainty and time-varying risk aversion. Our results provide support for models that feature both time variation in risk prices and time-varying risk as in, for example, Bekaert et al. (2009) and Creal and Wu (2018). However, our statistical measure of expected bond returns contrasts with recent survey-based measures like the one proposed by Buraschi et al. (2019), which is mostly related to financial (specifically, bond) volatility.

From a pure machine learning perspective in the context of asset pricing, we make three contributions. First, we find that NNs perform well even when, in the context of bond return regressions, we employ just yield-based variables (i.e., in a low dimensional setting). This finding emphasizes that the success of NNs is largely due to their ability to capture complex non-linearities in the data. Second, we document that non-linearities within macroeconomic categories (output, inflation, labor market, etc) are more important than interactions across categories. Finally, we document that a carefully chosen structure of the network (like group ensembling) may compensate for the depth of the network.

Overall machine learning methods that dispense with the linearity assumption in the return-predicting function may prove useful to improve our empirical understanding of asset price movements.

References

Ahn, Dong Hyun, Robert F. Dittmar, and Andrew Ronald Gallant (2002) "Quadratic Term Structure Models: Theory and Evidence," *Review of Financial Studies*, Vol. 15, No. 1, pp. 243–288, 1.

Arlot, Sylvain, Alain Celisse et al. (2010) "A survey of cross-validation procedures for model selection," *Statistics surveys*, Vol. 4, pp. 40–79.

Atanasov, Victoria, Stig Vinther Moller, and Richard Priestley (2019) "Consumption Fluctuations and Expected Returns," *The Journal of Finance*, Vol. n/a, No. n/a.

Bai, Jushan and Serena Ng (2003) "Determining the Number of Factors in Approximate Factor Models," *Econometrica*, Vol. 70, No. 1, pp. 191–221.

——— (2006) "Confidence Intervals for Diffusion Index Forecasts and Inference for Factor-Augmented Regressions," *Econometrica*, Vol. 74, No. 4, pp. 1133–1150.

——— (2008) "Forecasting economic time series using targeted predictors," *Journal of Econometrics*, Vol. 146, No. 2, pp. 304–317, October.

Bansal, Ravi and Ivan Shaliastovich (2013) "A Long-Run Risks Explanation of Predictability Puzzles in Bond and Currency Markets," *The Review of Financial Studies*, Vol. 26, No. 1, pp. 1–33, 01.

Bates, John M and Clive WJ Granger (1969) "The combination of forecasts," *Journal of the Operational Research Society*, Vol. 20, No. 4, pp. 451–468.

Bauer, Michael D. and James D. Hamilton (2018) "Robust Bond Risk Premia," *The Review of Financial Studies*, Vol. 31, No. 2, pp. 399–448, 09.

Bauer, Michael D. and Glenn D. Rudebusch (2017) "Resolving the Spanning Puzzle in Macro-Finance Term Structure Models," *Review of Finance*, Vol. 21, No. 2, pp. 511–553, 09.

Bekaert, Geert, Eric Engstrom, and Yuhang Xing (2009) "Risk, uncertainty, and asset prices," *Journal of Financial Economics*, Vol. 91, No. 1, pp. 59–82.

Bekaert, Geert, Eric C. Engstrom, and Nancy R. Xu (2019) "The Time Variation in Risk Appetite and Uncertainty," NBER Working Papers 25673, National Bureau of Economic Research, Inc.

Bishop, Christopher M (1995) "Regularization and complexity control in feed-forward networks."

Black, Fischer (1995) "Interest Rates as Options," *The Journal of Finance*, Vol. 50, No. 5, pp. 1371–1376.

Boivin, Jean and Serena Ng (2006a) "Are more data always better for factor analysis?" *Journal of Econometrics*, Vol. 132, No. 1, pp. 169–194.

——— (2006b) "Are more data always better for factor analysis?" *Journal of Econometrics*, Vol. 132, No. 1, pp. 169–194.

Breiman, Leo (2001) "Random forests," *Machine learning*, Vol. 45, No. 1, pp. 5–32.

Breiman, Leo, Jerome Friedman, Charles J. Stone, and R.A. Olshen (1984) *Classification and Regression Trees*: Taylor & Francis.

Buraschi, Andrea and Alexei Jiltsov (2007) "Habit Formation and Macroeconomic Models of the Term Structure of Interest Rates," *The Journal of Finance*, Vol. 62, No. 6, pp. 3009–3063.

Buraschi, Andrea, Ilaria Piatti, and Paul Whelan (2019) "Subjective Bond Risk Premia and Belief Aggregation,"Technical report.

Burns, Arthur F. and Wesley C. Mitchell (1946) *Measuring Business Cycles*: National Bureau of Economic Research, Inc.

Campbell, John Y and John H Cochrane (1999) "By force of habit: A consumption-based explanation of aggregate stock market behavior," *Journal of political Economy*, Vol. 107, No. 2, pp. 205–251.

Campbell, John Y. and Robert J. Shiller (1991) "Yield Spreads and Interest Rate Movements: A Bird's Eye View," *Review of Economic Studies*, Vol. 58, No. 3, pp. 495–514.

Campbell, John Y and Samuel B Thompson (2007) "Predicting excess stock returns out of sample: Can anything beat the historical average?" *The Review of Financial Studies*, Vol. 21, No. 4, pp. 1509–1531.

Campbell, John Y and Luis M Viceira (1999) "Consumption and portfolio decisions when expected returns are time varying," *The Quarterly Journal of Economics*, Vol. 114, No. 2, pp. 433–495.

——— (2004) "Long-horizon mean-variance analysis: A user guide," *Manuscript, Harvard University, Cambridge, MA*.

Chen, Hui, Winston Dou, and Leonid Kogan (2017) "Measuring the'Dark Matter'in Asset Pricing Models," *Working Paper*.

Chen, Luyang, Markus Pelger, and Jason Zhu (2019) "Deep learning in asset pricing," *Working Paper*.

Cieslak, Anna and Pavol Povala (2015) "Expected Returns in Treasury Bonds," *Review of Financial Studies*, Vol. 28, No. 10, pp. 2859–2901.

Clark, Todd E and Kenneth D West (2007) "Approximately normal tests for equal predictive accuracy in nested models," *Journal of econometrics*, Vol. 138, No. 1, pp. 291–311.

Clemen, Robert T (1989) "Combining forecasts: A review and annotated bibliography," *International journal of forecasting*, Vol. 5, No. 4, pp. 559–583.

Cochrane, John H. and Monika Piazzesi (2005) "Bond Risk Premia," *American Economic Review*, Vol. 95, No. 1, pp. 138–160, March.

Cooper, Ilan and Richard Priestley (2009) "Time-Varying Risk Premiums and the Output Gap," *The Review of Financial Studies*, Vol. 22, No. 7, pp. 2801–2833.

Coroneo, Laura, Domenico Giannone, and Michele Modugno (2016) "Unspanned Macroeconomic Factors in the Yield Curve," *Journal of Business & Economic Statistics*, Vol. 34, No. 3, pp. 472–485.

Creal, Drew D. and Jing Cynthia Wu (2018) "Bond Risk Premia in Consumption-based Models," NBER Working Papers 22183, National Bureau of Economic Research, Inc.

Cybenko, George (1989) "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, Vol. 2, No. 4, pp. 303–314.

Dai, Qiang, Kenneth J. Singleton, and Wei Yang (2007) "Regime Shifts in a Dynamic Term Structure Model of U.S. Treasury Bond Yields," *Review of Financial Studies*, Vol. 20, No. 5, pp. 1669–1706, 2007 12.

Dangl, Thomas and Michael Halling (2012) "Predictive regressions with time-varying coefficients," *Journal of Financial Economics*, Vol. 106, No. 1, pp. 157–181.

De Mol, Christine, Domenico Giannone, and Lucrezia Reichlin (2008) "Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components?" *Journal of Econometrics*, Vol. 146, No. 2, pp. 318–328.

Della Corte, Pasquale, Lucio Sarno, and Daniel Thornton (2008) "The expectation hypothesis of the term structure of very short-term rates: Statistical tests and economic value," *Journal of Financial Economics*, Vol. 89, No. 1, pp. 158–174.

Diaconis, P. and M. Shahshahani (1984) "On Nonlinear Functions of Linear Combinations," *SIAM Journal on Scientific and Statistical Computing*, Vol. 5, No. 1, pp. 175–191.

Diebold, Francis X and Robert S Mariano (1995) "Comparing predictive accuracy," *Journal of Business & economic statistics*, Vol. 20, pp. 134–144.

Dietterich, Thomas G (2000) "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15.

Dimopoulos, Yannis, Paul Bourret, and Sovan Lek (1995) "Use of some sensitivity criteria for choosing networks with good generalization ability," *Neural Processing Letters*, Vol. 2, No. 6, pp. 1–4.

Duffee, Gregory R. (2002) "Term Premia and Interest Rate Forecasts in Affine Models," *The Journal of Finance*, Vol. 57, No. 1, pp. 405–443.

Duffee, Greg (2011a) "Forecasting with the term structure: The role of no-arbitrage restrictions," Economics Working Paper Archive 576, The Johns Hopkins University,Department of Economics.

Duffee, Gregory R. (2011b) "Information in (and not in) the Term Structure," *Review of Financial Studies*, Vol. 24, No. 9, pp. 2895–2934.

Duffee, Gregory (2013) *Forecasting Interest Rates*, Vol. 2 of Handbook of Economic Forecasting, Chap. 0, pp. 385–426: Elsevier.

Dumas, Bernard, Alexander Kurshev, and Raman Uppal (2009) "Equilibrium Portfolio Strategies in the Presence of Sentiment Risk and Excess Volatility," *Journal of Finance*, Vol. 64, No. 2, pp. 579–629.

Elliott, Graham and Allan Timmermann (2004) "Optimal forecast combinations under general loss functions and forecast error distributions," *Journal of Econometrics*, Vol. 122, No. 1, pp. 47–79.

Fama, Eugene F and Robert R Bliss (1987) "The information in long-maturity forward rates," *The American Economic Review*, pp. 680–692.

Feldhutter, Peter, Christian Heyerdahl-Larsen, and Philipp Illeditsch (2016) "Risk Premia and Volatilities in a Nonlinear Term Structure Model," *Review of Finance*, Vol. 22, No. 1, pp. 337–380, 10.

Feng, Guanhao, Stefano Giglio, and Dacheng Xiu (2019) "Taming the Factor Zoo: A Test of New Factors," NBER Working Papers 25481, National Bureau of Economic Research, Inc.

Feng, Guanhao, Jingyu He, and Nicholas G Polson (2018) "Deep Learning for Predicting Asset Returns," *arXiv preprint arXiv:1804.09314*.

Feng, Guanhao, Nicholas Polson, and Jianeng Xu (2019) "Deep Learning Alpha," Chicago Booth Research Paper 23527, Chicago Booth.

Fleming, Jeff, Chris Kirby, and Barbara Ostdiek (2001) "The Economic Value of Volatility Timing," *Journal of Finance*, Vol. 56, No. 1, pp. 329–352, February.

Forni, Mario and Lucrezia Reichlin (1996) "Dynamic Common Factors in Large Cross-Sections," *Empirical Economics*, Vol. 21, No. 1, pp. 27–42.

——— (1998) "Let's Get Real: A Factor Analytical Approach to Disaggregated Business Cycle Dynamics," *The Review of Economic Studies*, Vol. 65, No. 3, pp. 453–473.

Frank, LLdiko E and Jerome H Friedman (1993) "A statistical view of some chemometrics regression tools," *Technometrics*, Vol. 35, No. 2, pp. 109–135.

Freyberger, Joachim, Andreas Neuhierl, and Michael Weber (2017) "Dissecting Characteristics Nonparametrically," CESifo Working Paper Series 6391, CESifo Group Munich.

Friedman, Jerome H (2001) "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001) *The elements of statistical learning*, Vol. 1: Springer series in statistics New York, NY, USA:.

Friedman, Jerome, Trevor Hastie, and Rob Tibshirani (2010) "Regularization paths for generalized linear models via coordinate descent," *Journal of statistical software*, Vol. 33, No. 1, p. 1.

Friedman, Jerome, Trevor Hastie, Holger Höfling, and Rob Tibshirani (2007) "Pathwise Coordinate Optimization," *The Annals of Applied Statistics*, Vol. 1, No. 2, pp. 302–332.

Fuster, Andreas, Paul Goldsmith-Pinkham, Tarun Ramadorai, and Ansgar Walther (2018) "Predictably unequal? the effects of machine learning on credit markets," *The Effects of Machine Learning on Credit Markets (November 6, 2018)*.

Gargano, Antonio, Davide Pettenuzzo, and Allan Timmermann (2019) "Bond Return Predictability: Economic Value and Links to the Macroeconomy," *Management Science*, Vol. 65, No. 2, pp. 508–540.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel (2006) "Extremely randomized trees," *Machine learning*, Vol. 63, No. 1, pp. 3–42.

Geweke, J. (1977) *The Dynamic Factor Analysis of Economic Time Series*: D.J Aigner and A.S. Goldberger, eds. (North-Holland, Amsterdam).

Geweke, John (2001a) "A note on some limitations of CRRA utility," *Economics Letters*, Vol. 71, No. 3, pp. 341 – 345.

——— (2001b) "A note on some limitations of CRRA utility," *Economics letters*, Vol. 71, No. 3, pp. 341–345.

Giacoletti, Marco, Kristoffer Laursen, and Kenneth J Singleton (2016) "Learning, dispersion of beliefs, and risk premiums in an arbitrage-free term structure model," *Working Paper*.

Giannone, Domenico, Michele Lenza, and Giorgio Primiceri (2017) "Economic predictions with big data: The illusion of sparsity," *Working Paper*.

Giglio, Stefano and Dacheng Xiu (2017) "Inference on Risk Premia in the Presence of Omitted Factors," NBER Working Papers 23527, National Bureau of Economic Research, Inc.

Goodfellow, Ian, Yoshua Bengio, Aaron Courville, and Yoshua Bengio (2016) *Deep learning*, Vol. 1: MIT press Cambridge.

Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu (2018) "Empirical Asset Pricing via Machine Learning," Chicago Booth Research Paper 18-04, Chicago Booth.

Gurkaynak, Refet S., Brian Sack, and Jonathan H. Wright (2007) "The U.S. Treasury yield curve: 1961 to the present," *Journal of Monetary Economics*, Vol. 54, No. 8, pp. 2291–2304, November.

Hansen, Lars Kai and Peter Salamon (1990) "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 12, No. 10, pp. 993–1001.

Harvey, David, Stephen Leybourne, and Paul Newbold (1997) "Testing the equality of prediction mean squared errors," *International Journal of forecasting*, Vol. 13, No. 2, pp. 281–291.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015a) "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

——— (2015b) "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

Heaton, J. B., N. G. Polson, and J. H. Witte (2017) "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, Vol. 33, No. 1, pp. 3–12.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989) "Multilayer feedforward networks are universal approximators," *Neural networks*, Vol. 2, No. 5, pp. 359–366.

Huang, Jing-Zhi and Zhan Shi (2019) "Determinants of Bond Risk Premia: A Machine-Learning-Based Resolution of the Spanning Controversy," Working Papers 2019-12, Penn State.

Ioffe, Sergey and Christian Szegedy (2015) "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*.

Johannes, Michael, Arthur Korteweg, and Nicholas Polson (2014) "Sequential Learning, Predictability, and Optimal Portfolio Returns," *Journal of Finance*, Vol. 69, No. 2, pp. 611–644, April.

Joslin, Scott, Marcel Priebsch, and Kenneth J. Singleton (2014) "Risk Premiums in Dynamic Term Structure Models with Unspanned Macro Risks," *The Journal of Finance*, Vol. 69, No. 3, pp. 1197–1233.

Kandel, Shmuel and Robert F. Stambaugh (1996) "On the Predictability of Stock Returns: An Asset-Allocation Perspective," *The Journal of Finance*, Vol. 51, No. 2, pp. 385–424.

Kelly, Bryan and Seth Pruitt (2013) "Market Expectations in the Cross-Section of Present Values," *The Journal of Finance*, Vol. 68, No. 5, pp. 1721–1756.

——— (2015) "The three-pass regression filter: A new approach to forecasting using many predictors," *Journal of Econometrics*, Vol. 186, No. 2, pp. 294–316.

Kelly, Bryan, Seth Pruitt, and Yinan Su (2018) "Characteristics Are Covariances: A Unified Model of Risk and Return," NBER Working Papers 24540, National Bureau of Economic Research, Inc.

Kolmogorov, A. K. (1957) "On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition," *Doklady Akademii Nauk SSSR*, Vol. 114, pp. 369–373.

Kozak, Serhiy, Stefan Nagel, and Shrihari Santosh (2017) "Shrinking the Cross Section," NBER Working Papers 24070, National Bureau of Economic Research, Inc.

Lazarus, Eben, Daniel J. Lewis, James H. Stock, and Mark W. Watson (2018) "HAR Inference: Recommendations for Practice," *Journal of Business & Economic Statistics*, Vol. 36, No. 4, pp. 541–559, October.

Le, Anh and Kenneth J. Singleton (2013) "The Structure of Risks in Equilibrium Affine Models of Bond Yields," working paper, Stanford Business School WP.

Le, Anh, Kenneth J. Singleton, and Qiang Dai (2010) "Discrete-Time AffineQ Term Structure Models with Generalized Market Prices of Risk," *The Review of Financial Studies*, Vol. 23, No. 5, pp. 2184–2227, 03.

Leippold, Markus and Liuren Wu (2003) "Design and Estimation of Quadratic Term Structure Models," *Review of Finance*, Vol. 7, No. 1, pp. 47–73.

Lek, Sovan, Alain Belaud, Ioannis Dimopoulos, J Lauga, and J Moreau (1995) "Improved estimation, using neural networks, of the food consumption of fish populations," *Marine and Freshwater Research*, Vol. 46, No. 8, pp. 1229–1236.

Lek, Sovan, Marc Delacoste, Philippe Baran, Ioannis Dimopoulos, Jacques Lauga, and Stephane Aulagnier (1996) "Application of neural networks to modelling nonlinear relationships in ecology," *Ecological modelling*, Vol. 90, No. 1, pp. 39–52.

Liu, Yan and Jing Cynthia Wu (2019) "Reconstructing the Yield Curve," NBER Working Paper 24070, University of Notre Dame.

Ludvigson, Sydney C. and Serena Ng (2009) "Macro Factors in Bond Risk Premia," *Review of Financial Studies*, Vol. 22, No. 12, pp. 5027–5067, December.

McCracken, Michael W. and Serena Ng (2015) "FRED-MD: A Monthly Database for Macroeconomic Research," Working Papers 2015-12, Federal Reserve Bank of St. Louis.

McCracken, Michael W and Serena Ng (2016) "FRED-MD: A monthly database for macroeconomic research," *Journal of Business & Economic Statistics*, Vol. 34, No. 4, pp. 574–589.

Messmer, Marcial (2017) "Deep Learning and the Cross-Section of Expected Returns."

Mullainathan, Sendhil and Jann Spiess (2017) "Machine learning: an applied econometric approach," *Journal of Economic Perspectives*, Vol. 31, No. 2, pp. 87–106.

Nesterov, Yuri (1983) "A method for solving the convex programming problem with convergence rate $O(1/k\hat{2})$," *Dokl. Akad. Nauk SSSR*, Vol. 269, pp. 543–547.

Newbold, Paul and Clive WJ Granger (1974) "Experience with forecasting univariate time series and the combination of forecasts," *Journal of the Royal Statistical Society: Series A (General)*, Vol. 137, No. 2, pp. 131–146.

Newey, Whitney K. and Kenneth D. West (1987) "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix," *Econometrica*, Vol. 55, No. 3, pp. 703–708.

Pastor, Lubos and Pietro Veronesi (2005) "Rational IPO Waves," *The Journal of Finance*, Vol. 60, No. 4, pp. 1713–1757.

Piazzesi, Monika, Juliana Salomao, and Martin Schneider (2015) "Trend and cycle in bond premia,"Technical report.

Polson, Nicholas G. and Vadim Sokolov (2017) "Deep Learning: A Bayesian Perspective," *Bayesian Analysis*, Vol. 12, No. 4, pp. 1275–1304, 12.

Rapach, David and Guofu Zhou (2019) "Sparse Macro Factors," ssrn working paper.

Rapach, David E, Jack K Strauss, and Guofu Zhou (2010) "Out-of-sample equity premium prediction: Combination forecasts and links to the real economy," *The Review of Financial Studies*, Vol. 23, No. 2, pp. 821–862.

Rapach, David E., Jack K. Strauss, and Guofu Zhou (2013) "International Stock Return Predictability: What Is the Role of the United States?" *Journal of Finance*, Vol. 68, No. 4, pp. 1633–1662.

Ripley, Brian D (1994) "Neural networks and related methods for classification," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 409–456.

Rossi, A.G. (2018) "Predicting Stock Market Returns with Machine Learning,"Technical report, Working paper.

Sargent, Thomas and Christopher Sims (1977) "Business cycle modeling without pretending to have too much a priori economic theory," Working Papers 55, Federal Reserve Bank of Minneapolis.

Sarno, Lucio, Paul Schneider, and Christian Wagner (2016) "The economic value of predicting bond risk premia," *Journal of Empirical Finance*, Vol. 37, pp. 247–267.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller (1998) "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, Vol. 10, No. 5, pp. 1299–1319.

Sirignano, Justin A., Apaar Sadhwani, and Kay Giesecke (2018) "Deep Learning for Mortgage Risk," economics working paper archive, Stanford Working Paper.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014) "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958.

Stock, James H and Mark W Watson (1998) "A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series,"Technical report, National Bureau of Economic Research.

——— (2002a) "Macroeconomic Forecasting Using Diffusion Indexes," *Journal of Business & Economic Statistics*, Vol. 20, No. 2, pp. 147–162, April.

——— (2002b) "Forecasting Using Principal Components From a Large Number of Predictors," *Journal of the American Statistical Association*, Vol. 97, pp. 1167–1179, December.

Stock, James H. and Mark Watson (2006) "Forecasting with Many Predictors," Vol. 1: Elsevier, 1st edition, Chap. 10, pp. 515–554.

Stone, Mervyn and Rodney J Brooks (1990) "Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 237–269.

Sung, AH (1998) "Ranking importance of input parameters of neural networks," *Expert Systems with Applications*, Vol. 15, No. 3-4, pp. 405–411.

Thornton, Daniel L and Giorgio Valente (2012) "Out-of-sample predictions of bond excess returns and forward rates: An asset allocation perspective," *The Review of Financial Studies*, Vol. 25, No. 10, pp. 3141–3168.

Veronesi, Pietro (2004) "Belief-dependent Utilities, Aversion to State-Uncertainty and Asset Prices,"," crsp working papers, Center for Research in Security Prices, Graduate School of Business, University of Chicago.

Wachter, Jessica A. (2006) "A consumption-based model of the term structure of interest rates," *Journal of Financial Economics*, Vol. 79, No. 2, pp. 365–399, February.

Welch, Ivo and Amit Goyal (2008) "A comprehensive look at the empirical performance of equity premium prediction," *The Review of Financial Studies*, Vol. 21, No. 4, pp. 1455–1508.

Werbos, Paul J. (1974) "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences," Ph.D. dissertation, Harvard University.

——— (1982) *Applications of advances in nonlinear sensitivity analysis*, pp. 762–770: Springer.

——— (1988) "Generalization of backpropagation with application to a recurrent gas market model," *Neural networks*, Vol. 1, No. 4, pp. 339–356.

Wu, Jing Cynthia and Fan Dora Xia (2016) "Measuring the Macroeconomic Impact of Monetary Policy at the Zero Lower Bound," *Journal of Money, Credit and Banking*, Vol. 48, No. 2-3, pp. 253–291.

Wu, Tong Tong, Kenneth Lange et al. (2008) "Coordinate descent algorithms for lasso penalized regression," *The Annals of Applied Statistics*, Vol. 2, No. 1, pp. 224–244.

Yang, Wei (2011) "Long-run risk in durable consumption," *Journal of Financial Economics*, Vol. 102, No. 1, pp. 45–61.

Zou, Hui and Trevor Hastie (2005) "Method of Conjugate Gradients for Solving Linear Systems," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 67, No. 2, pp. 301–320.

TABLE 1: **Forecasting Annual Holding-period Returns with Forward Rates**

This table reports the out-of-sample R^2_{oos} obtained using forward rates to predict annual excess bond returns for different maturities and across methodologies. To compute the out-of-sample R^2_{oos} we compare the forecasts obtained from each methodology to the expectation hypothesis (i.e., to the prediction based on the historical mean). In addition to the R^2_{oos} we report the p-value for the null hypothesis $R^2_{oos} \leq 0$ calculated as in Clark and West (2007). Notice that we report a p-value only when the R^2_{oos} is positive. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Models	R^2_{oos}						p-value						$R2_{oos}$ EW	p-value EW
	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(EW)}_{t+1}$	$xr^{(EW)}_{t+1}$
Panel A: PCA and PLS														
PCA (10 components)	-53.0%	-36.2%	-27.5%	-20.3%	-12.4%	-0.7%							-17.7%	
PCA (5 components)	-54.9%	-38.8%	-30.4%	-20.3%	-15.3%	-3.1%							-20.0%	
PCA (3 components)	-17.6%	-10.2%	-5.3%	1.0%	2.9%	10.2%				0.052	0.028	0.008	1.0%	0.036
PCA-Squared (5 Components)	-55.0%	-42.1%	-32.5%	-22.6%	-16.4%	-5.5%							-22.4%	
PCA-Squared (3 Components)	-46.8%	-38.8%	-30.6%	-21.0%	-17.0%	-8.9%							-22.5%	
Partial Least Squares (5 components)	-56.3%	-40.5%	-33.4%	-25.7%	-18.6%	-9.5%							-24.8%	
Partial Least Squares (3 components)	-57.9%	-40.7%	-31.9%	-22.9%	-14.6%	-1.8%							-20.2%	
Panel B: Penalized Linear Regressions														
Ridge	-40.1%	-25.8%	-18.9%	-11.4%	-6.0%	4.6%						0.011	-9.9%	
Lasso	-11.5%	-8.1%	-2.5%	0.2%	2.3%	9.3%				0.072	0.044	0.010	2.3%	0.041
Elastic Net	-10.7%	-8.4%	-5.1%	0.4%	0.3%	7.0%				0.064	0.060	0.018	0.7%	0.052
Panel C: Regression Trees and Neural Networks														
Gradient Boosted Tree	4.9%	4.9%	7.0%	10.8%	7.1%	11.2%	0.002	0.007	0.004	0.001	0.003	0.000	8.9%	0.003
Random Forest	12.2%	13.1%	15.2%	17.4%	14.9%	16.0%	0.000	0.000	0.000	0.000	0.000	0.000	15.9%	0.000
Extreme Tree	3.9%	10.1%	14.3%	16.9%	19.7%	24.6%	0.014	0.009	0.004	0.001	0.001	0.000	17.9%	0.001
NN - 1 Layer (3 nodes)	12.7%	16.4%	19.5%	21.6%	23.1%	26.4%	0.028	0.014	0.007	0.003	0.002	0.001	23.0%	0.002
NN - 1 Layer (5 nodes)	7.0%	10.7%	14.6%	17.2%	18.9%	22.9%	0.023	0.013	0.008	0.004	0.003	0.002	18.2%	0.003
NN - 1 Layer (7 nodes)	-3.2%	3.0%	7.1%	12.1%	13.6%	17.1%		0.026	0.009	0.003	0.004	0.004	11.5%	0.004
NN - 2 Layer (3 nodes each)	7.4%	10.8%	12.9%	14.7%	16.3%	19.3%	0.053	0.031	0.019	0.012	0.008	0.004	16.4%	0.009
NN - 2 Layer (5 nodes each)	8.9%	13.0%	16.0%	18.5%	20.4%	23.6%	0.027	0.013	0.007	0.004	0.003	0.002	20.2%	0.003
NN - 2 Layer (7 nodes each)	9.0%	15.2%	18.1%	21.2%	23.7%	27.4%	0.013	0.004	0.002	0.001	0.001	0.001	23.1%	0.001
NN - 3 Layer (3 nodes each)	3.4%	8.9%	10.9%	11.2%	12.7%	15.1%	0.095	0.040	0.026	0.017	0.012	0.006	12.9%	0.013
NN - 3 Layer (5 nodes each)	3.6%	7.4%	8.9%	10.6%	11.6%	13.7%	0.118	0.070	0.057	0.037	0.028	0.018	11.7%	0.031
NN - 3 Layer (7 nodes each)	3.9%	6.7%	8.3%	8.9%	11.1%	13.2%	0.088	0.061	0.043	0.031	0.023	0.014	10.8%	0.025
NN - 3 Layer (5,4,3 nodes each)	-0.9%	2.8%	5.6%	6.5%	8.3%	10.6%		0.139	0.079	0.061	0.040	0.020	8.0%	0.044
NN - 1 Layer (7 nodes), lagged inputs: $t-1:t-11$	6.4%	14.4%	17.0%	19.5%	20.3%	24.4%	0.028	0.009	0.006	0.003	0.001	0.001	20.6%	0.002

TABLE 2: **Forecasting Annual Holding-period Returns with Forward Rates and Macroeconomic Variables**

This table reports the out-of-sample R^2_{oos} obtained using forward rates and a large panel of macroeconomic variables to predict annual excess bond returns for different maturities. To compute the out-of-sample R^2_{oos} we compare the forecasts obtained from each methodology to the expectation hypothesis (i.e., prediction based on the historical mean). In addition to the R^2_{oos} we report the p-value for the null hypothesis $R^2_{oos} \leq 0$ calculated as in Clark and West (2007). Notice that we report a p-value only when the R^2_{oos} is positive. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12. Penalized regressions are estimated including macro-economic variables plus either raw forward rates or a linear combination of forward rates as introduced by Cochrane and Piazzesi (2005) (CP). Similarly, neural networks are estimated either adding the CP factor as an additional regressor in the output layer ("fwd rates direct") or by estimating a separate network for forward rates and ensembling both macro and forward rates networks in the output layer ("fwd rates net").

Models	R^2_{oos}						p-value						$R2_{oos}$ EW	p-value EW
	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(EW)}_{t+1}$	$xr^{(EW)}_{t+1}$
Panel A: PCA and PLS														
PCA - first 8 PC'S	-9.8%	-2.9%	0.3%	3.0%	3.3%	4.5%			0.004	0.004	0.003	0.002	1.8%	0.003
PCA as in Ludvigson and Ng (2009)	-3.4%	0.2%	1.6%	1.6%	-1.4%	-4.7%		0.007	0.006	0.007			-1.3%	
PLS - 8 components	-40.7%	-19.7%	-12.0%	-8.2%	-2.7%	3.4%						0.000	-6.4%	
Panel B: Penalized Linear Regressions														
Ridge (using CP factor)	-45.3%	-23.6%	-16.7%	-13.2%	-3.1%	5.3%						0.000	-5.6%	
Lasso (using CP factor)	6.4%	11.2%	12.9%	14.4%	19.6%	23.7%	0.008	0.004	0.002	0.002	0.001	0.000	21.0%	0.001
Elastic Net (using CP factor)	6.4%	11.0%	14.3%	15.7%	21.7%	29.1%	0.007	0.004	0.002	0.001	0.001	0.000	22.0%	0.001
Ridge (using fwd rates directly)	-52.2%	-28.7%	-22.7%	-18.3%	-13.1%	-3.5%							-15.4%	
Lasso (using fwd rates directly)	11.0%	12.0%	12.3%	16.4%	19.9%	23.6%	0.003	0.003	0.002	0.006	0.007	0.004	20.7%	0.004
Elastic Net (using fwd rates directly)	10.2%	14.2%	16.0%	13.2%	19.9%	23.6%	0.005	0.003	0.003	0.003	0.002	0.002	21.0%	0.002
Panel C: Regression Trees and Neural Networks														
Gradient Boosted Tree	13.1%	15.9%	18.1%	23.8%	22.5%	25.5%	0.004	0.006	0.005	0.003	0.004	0.001	26.2%	0.002
Random Forest	26.7%	21.5%	22.0%	24.4%	20.0%	25.0%	0.003	0.003	0.002	0.001	0.004	0.002	26.6%	0.002
Extreme Tree	23.0%	23.4%	22.3%	23.7%	29.9%	29.6%	0.004	0.005	0.005	0.004	0.001	0.003	29.2%	0.002
NN 1 Layer (32 nodes), fwd rates direct	6.0%	13.6%	17.8%	22.0%	22.5%	26.5%	0.003	0.001	0.000	0.000	0.000	0.000	22.4%	0.000
NN 2 Layer (32, 16 nodes), fwd rates direct	16.6%	21.5%	24.9%	28.0%	29.4%	32.3%	0.001	0.000	0.000	0.000	0.000	0.000	29.3%	0.000
NN 3 Layer (32, 16, 8 nodes), fwd rates direct	24.8%	26.3%	29.7%	32.0%	31.7%	33.7%	0.000	0.000	0.000	0.000	0.000	0.000	32.5%	0.000
NN 1 Layer (32 nodes), fwd rates net (1 layer: 3 nodes)	8.4%	19.0%	23.8%	25.6%	27.2%	29.5%	0.003	0.001	0.001	0.001	0.001	0.000	27.2%	0.001
NN 2 Layer (32,16, nodes), fwd rates net (1 layer: 3 nodes)	12.1%	15.7%	20.0%	23.5%	25.4%	28.1%	0.008	0.004	0.002	0.001	0.001	0.001	25.1%	0.001
NN 3 Layer (32,16, 8 nodes), fwd rates net (1 layer: 3 nodes)	7.6%	16.3%	20.2%	23.7%	25.0%	28.1%	0.014	0.005	0.004	0.002	0.002	0.001	25.0%	0.002
NN 1 Layer Group Ensem (1 node per group), fwd rates direct	12.6%	17.3%	21.6%	24.2%	25.9%	29.6%	0.002	0.001	0.001	0.000	0.000	0.000	25.9%	0.000
NN 1 Layer Group Ensem (1 node per group), fwd rates net (1 layer: 3 nodes)	20.0%	25.6%	29.5%	31.2%	33.6%	36.3%	0.002	0.001	0.000	0.000	0.000	0.000	34.0%	0.000
NN 2 Layer Group Ensem (2,1 nodes per group / hidden layer), fwd rates net (2 layer: 3 nodes)	17.3%	23.6%	27.8%	29.8%	31.0%	33.0%	0.006	0.001	0.001	0.000	0.000	0.000	31.6%	0.000
NN 3 Layer Group Ensem (3, 2, 1 nodes per group / hidden layer), fwd rates net (3 layer: 3 nodes)	13.6%	20.0%	23.7%	26.1%	27.5%	30.7%	0.011	0.005	0.003	0.002	0.002	0.001	27.9%	0.002

TABLE 3: **Forecasting Performances in Expansions and Recessions**

This table reports the out-of-sample performances, measured by R^2_{oos}, separately for expansions (Exp) and recessions (Rec) as defined by the NBER recession index. For ease of exposition, we report the results for the Principal Component Regression with three PCs and for the best performing non-linear methodologies, that is extreme trees and the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables – (see Table 1-2 for reference). We focus on the prediction exercise with two- and ten-year maturity bonds. We denote in boldface values for which the predictive accuracy of a given model is better than that obtained from the EH benchmark at the 5% level (p-value calculated as in Clark and West (2007)). The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

	Forward rates		Fwd rates + Macro	
	R^2_{oos}		R^2_{oos}	
	Exp	Rec	Exp	Rec
PCA (10-year maturity)	7.69%	**34.88%**	-1.08%	-41.19%
PCA (2-year maturity)	-15.27%	-30.04%	-8.30%	**22.98%**
Extreme tree (10-year maturity)	**26.80%**	2.85%	**33.38%**	-8.74%
Extreme tree (2-year maturity)	**7.19%**	-14.18%	**25.08%**	11.63%
Neural net (10-year maturity)	**26.28%**	**27.33%**	**35.70%**	**42.54%**
Neural net (2-year maturity)	**9.42%**	**30.31%**	**17.34%**	**34.23%**

TABLE 4: **Ex-post Diagnostics Based on Principal Components Forecasts**

This table reports the in-sample R^2 of a predictive regression where the dependent variable is the year-on-year change in the first three principal components extracted from the term structure of interest rates. The first row reports results when the independent variables are the lagged first three principal components. The second and third rows display the in-sample R^2 when, in addition to the first three principal components, we include the factors extracted from the best performing neural networks obtained using either forward rates only or forward rates plus a large set of macroeconomic variables (see Table 1-2 for reference).

	Level	Slope	Curvature
PCA	9.28%	21.66%	48.70%
Neural net (fwd rates only)	36.67%	22.05%	70.52%
Neural net (fwd rates + macro)	30.98%	30.91%	65.43%

TABLE 5: **Economic Significance of Bond Predictability**

This table reports the annualized certainty equivalent values (in %) for portfolio decisions based on the out-of-sample forecasts of bond excess returns for an investor with either mean-variance (Panel A) or power utility (Panel B) and a coefficient of risk aversion equal to five. The table reports two asset allocation exercises. In the univariate asset allocation case, the investor selects either the two- or the ten-year bond, along with the one-year short-rate. In the multivariate case, the investor selects bonds across the six maturities, two- to five-, seven- and ten-years. The asset allocation decision is based on the predictions implied by either the best performing regression tree specification, i.e., extreme tree, or the best performing neural network, namely the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables – (see Table 1-2 for reference). The row Δ reports the value added by NN relative to extreme tree within each application ("Fwd rates" and "Fwd + Macro"). The column Δ reports the value added by "Fwd+Macro" relative to "Fwd rates" within each model (NN and extreme tree). The models are benchmarked against the expectation hypothesis. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12. Statistical significance is based on a one-sided Diebold and Mariano (1995) test as extended by Harvey et al. (1997) to account for autocorrelation in the forecasting errors. We flag in bold those values that are statistically significant at the 5% confidence level.

Panel A: Mean-Variance Utility

	2-year maturity			10-year maturity			All		
	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ
Neural net	-0.044	-0.002	0.042	**2.622**	**4.194**	**1.571**	**3.555**	**5.015**	**1.461**
p-value	(0.552)	(0.981)	(0.422)	(0.002)	(0.000)	(0.000)	(0.017)	(0.050)	(0.000)
Extreme tree	-0.151	-0.022	0.128	**2.881**	**2.955**	0.074	3.266	**3.961**	0.695
p-value	(0.464)	(0.675)	(0.355)	(0.001)	(0.000)	(0.864)	(0.078)	(0.078)	(0.112)
Δ	0.106	0.020		-0.258	**1.239**		0.289	**1.054**	
p-value	(0.421)	(0.680)		(0.510)	(0.001)		(0.722)	(0.024)	

Panel B: Power Utility

	2-year maturity			10-year maturity			All		
	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ
Neural net	**0.056**	**0.111**	**0.054**	**2.714**	**3.077**	**0.363**	**3.152**	**4.829**	**1.678**
p-value	(0.002)	(0.000)	(0.000)	(0.000)	(0.000)	(0.020)	(0.000)	(0.000)	(0.000)
Extreme tree	0.022	**0.092**	**0.072**	**3.301**	**2.511**	**-0.795**	**3.831**	**3.943**	0.113
p-value	(0.486)	(0.002)	(0.000)	(0.000)	(0.000)	(0.021)	(0.000)	(0.000)	(0.835)
Δ	0.034	0.017		**-0.591**.	**0.567**		**-0.680**	**0.886**	
p-value	(0.212)	(0.267)		(0.023)	(0.024)		(0.030)	(0.029)	

TABLE 6: **Sharpe Ratios in Expansions and Recessions**

This table reports the out-of-sample annualized Sharpe ratio, separately for expansions (Exp) and recessions (Rec) as defined by the NBER recession index. We report the results for the benchmarking regression that employ the first three principal components of the yield curve, and for the best performing non-linear methodologies, that is extreme trees and the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables – (see Table 1-2 for reference). We focus on the prediction of two- and ten-year bond. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

	Forward rates		Fwd rates + Macro	
	Exp	Rec	Exp	Rec
PCA (10-year maturity)	0.087	1.364	-0.118	0.190
PCA (2-year maturity)	0.037	1.524	0.403	1.356
Extreme tree (10-year maturity)	0.261	0.521	0.491	0.555
Extreme tree (2-year maturity)	0.253	1.688	0.740	1.541
Neural net (10-year maturity)	0.506	1.769	0.749	1.707
Neural net (2-year maturity)	1.077	2.384	1.093	2.098

TABLE 7: **Drivers of Bond Risk Premia**

This table reports the regression estimates of realized (Panel A) and expected (Panel B and C) bond excess returns on 10-year bonds on a set of structural determinants of risk premia (see discussion in the paper for details). The expected bond return (dependent variable) is based on the predictions implied by the best performing neural network, namely the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables – (see Table 1-2 for reference). We standardize both left- and right-hand side variables, so that a 1-standard deviation change in the right hand variables implies a β-standard deviation in the dependent variable. We report the regression estimates as well as Newey-West p-values. Bold font indicates significance at the 5% level. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Panel A: 10-year realized bond excess returns

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	-0.19	0.29							6.43
	(0.41)	(0.19)							
(ii)			0.01						0.25
			(0.90)						
(iii)				0.01					0.25
				(0.81)					
(iv)					-0.14	0.28			5.82
					(0.42)	(0.14)			
(v)							**0.25**	-0.22	6.36
							(0.02)	(0.38)	
(vi)		0.27	0.04	-0.03	0.09				7.01
		(0.08)	(0.66)	(0.67)	(0.14)				

Panel B: 10-year expected bond excess returns (fwd rates)

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	0.02	0.31							3.06
	(0.96)	(0.22)							
(ii)			**0.35**						9.85
			(0.01)						
(iii)				0.15					1.58
				(0.21)					
(iv)					**0.40**	-0.11			10.34
					(0.00)	(0.72)			
(v)							-0.11	**0.74**	3.30
							(0.53)	(0.04)	
(vi)		0.01	**0.43**	0.04	**0.28**				23.69
		(0.95)	(0.00)	(0.77)	(0.00)				

Panel C: 10-year expected bond excess returns (fwd rates + macro)

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	-0.35	**0.55**							8.22
	(0.43)	(0.01)							
(ii)			**0.32**						11.80
			(0.02)						
(iii)				**0.27**					6.06
				(0.02)					
(iv)					**0.38**	-0.15			8.49
					(0.01)	(0.71)			
(v)							0.05	0.59	5.44
							(0.80)	(0.14)	
(vi)		0.19	**0.35**	0.15	0.21				25.22
		(0.35)	(0.00)	(0.14)	(0.09)				

TABLE 8: **Statistical Vs. Subjective Forecasts of Bond Risk Premia**

This table reports the correlation between our machine learning implied forecasts and existing measures of bond risk premia based on subjective forecasts or on asset pricing models with learning dynamics. Panel A: shows the results for the forecasts generated using only the forward rates, whereas Panel B: shows the results for the forecasts generated using both forward rates and a large panel of macroeconomic variables. Correlations are computed with respect to the subjective bond risk premia in Buraschi et al. (2019) (EBR^*_{10y} and EBR^*_{2y}), the subjective risk premia measure proposed by Piazzesi et al. (2015) (SUBJ_BRP), and the out-of-sample bond returns forecasts in Giacoletti et al. (2016) (GLS). In parentheses we report Newey-West p-values with 12 lags. Bold font indicates significance at the 5% level. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Panel A: Forecasting with forward rates

	10-year maturity		
	EBR^*_{10y}	SUBJ_BRP	GLS
Extreme tree	-7.5%	**49.5%**	**52.3%**
	(0.45)	(0.00)	(0.00)
NN - 1 Layer (3 nodes)	3.3%	**56.3%**	**59.5%**
	(0.75)	(0.00)	(0.00)

	2-year maturity		
	EBR^*_{2y}	SUBJ_BRP	GLS
Extreme tree	-18.7%	**42.7%**	**63.8%**
	(0.17)	(0.00)	(0.00)
NN - 1 Layer (3 nodes)	4.1%	**53.8%**	**50.3%**
	(0.78)	(0.00)	(0.00)

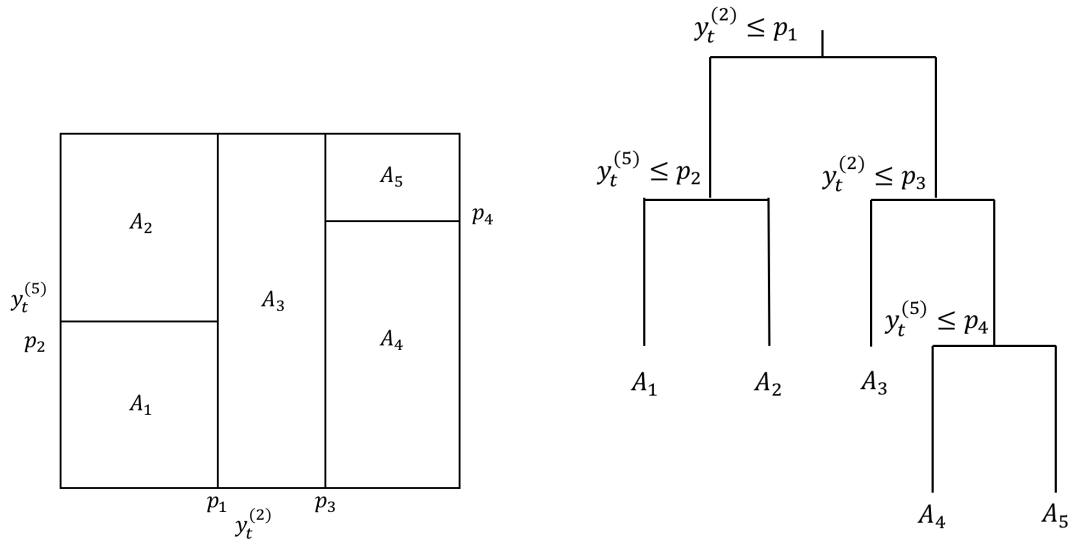
Panel B: Forecasting with forward rates and macro variables

	10-year maturity		
	EBR^*_{10y}	SUBJ_BRP	GLS
Extreme tree	-1.6%	**40.9%**	**48.3%**
	(0.88)	(0.00)	(0.00)
NN - 1 Layer Group Ensem + fwd rate net	13.4%	**38.0%**	**47.2%**
	(0.26)	(0.00)	(0.00)

	2-year maturity		
	EBR^*_{2y}	SUBJ_BRP	GLS
Extreme tree	3.5%	21.3%	20.0%
	(0.80)	(0.13)	(0.16)
NN - 1 Layer Group Ensem + fwd rate net	6.4%	**27.5%**	**30.3%**
	(0.61)	(0.03)	(0.03)

FIGURE 1: **Example of a Regression Tree**

This figure shows an example of a regression tree for a predictive regression with a univariate target variable, e.g., the holding period excess return of a one-year treasury bond, and two predictors, e.g., the two-year and the five-year forward rates, which we label $y_t^{(2)}$ and $y_t^{(5)}$. Left panel shows the partition of the two-dimensional regression space by recursive splitting. Right panel shows the corresponding regression tree.



(a) Recursive Binary Splitting (b) Partition Tree

FIGURE 2: **Examples of a Neural Network with only forward rates**

This figure shows a neural network with one hidden layer when forecasts are based only on forward rates as e.g. in Cochrane and Piazzesi (2005).

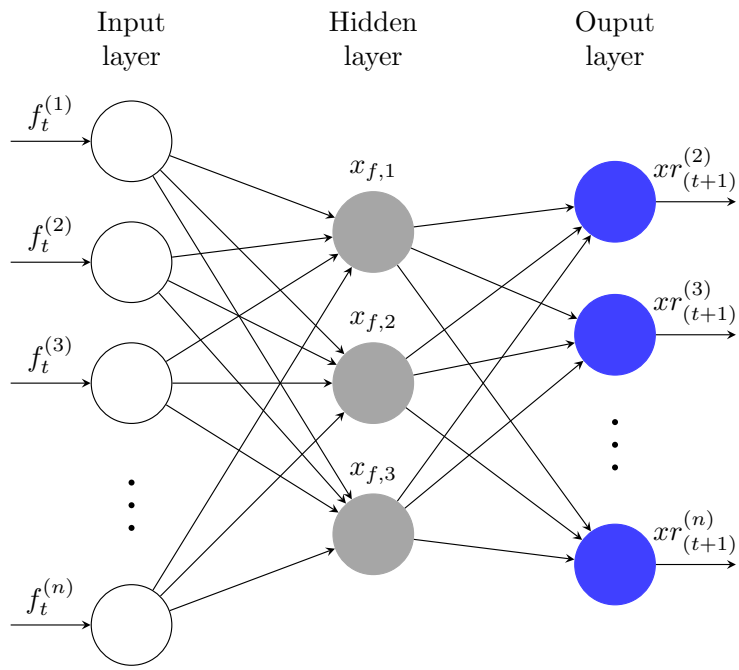
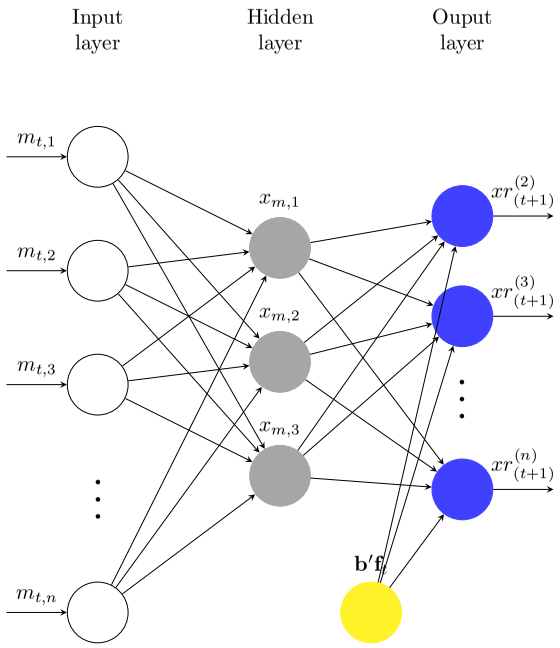
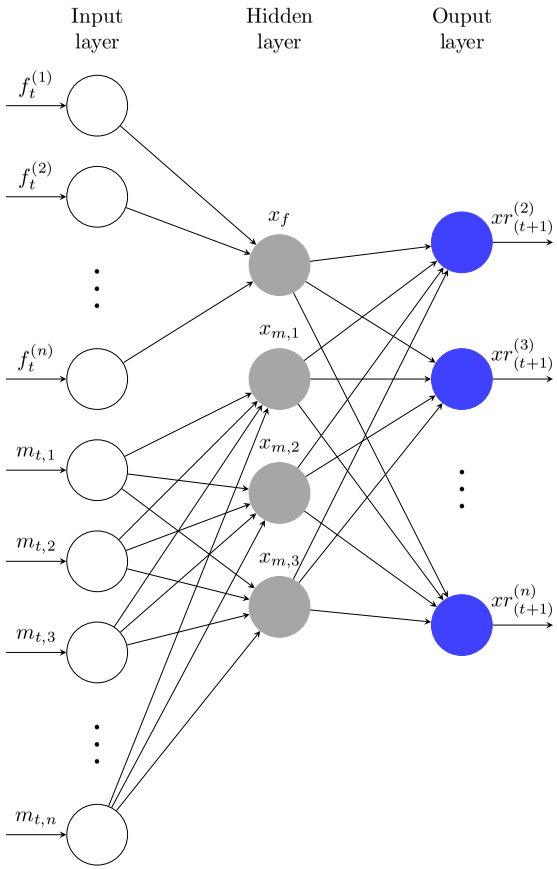


FIGURE 3: **Examples of Neural Networks with macro and forward rates**

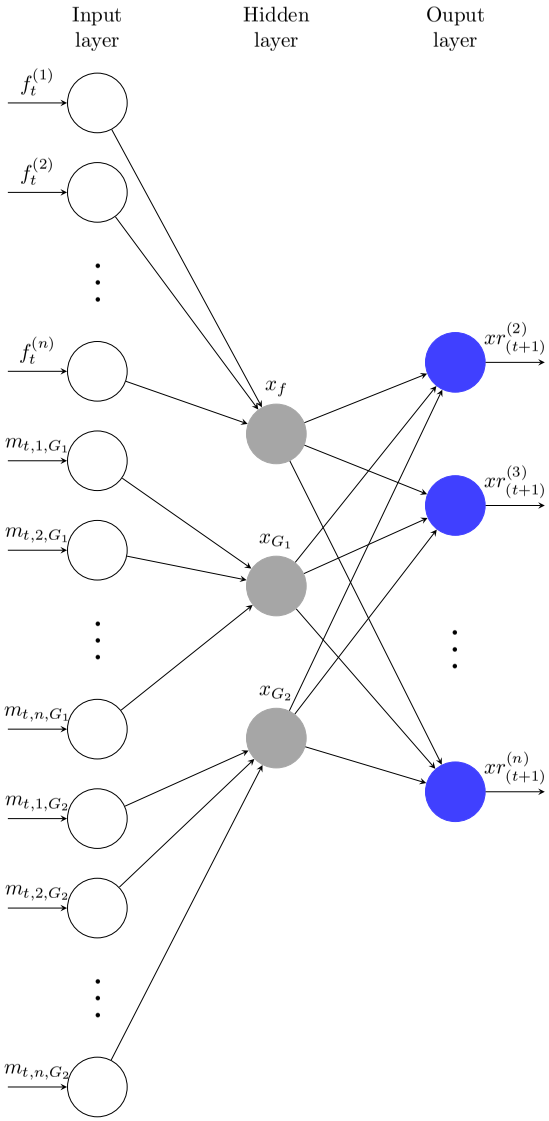
This figure shows examples of the network structures used in the paper. The left panel shows the neural network with a linear combination of forward rates, $b'f_t$, that is included as an exogenous regressors (in the paper such specification is called macro + fwd rates direct). This structure simulate the idea of Ludvigson and Ng (2009) in which the latent macro factors are extracted from a large cross-section of macroeconomic variables and forward rates are included as a linear combination as proposed by Cochrane and Piazzesi (2005). The center panel displays a network structure whereby macro variables, $m_{t,i}$, and forward rates, $f_t^{(n)}$, define two separate groups (in the paper such specification is called macro + fwd rates net). The right panel shows the group-ensemble network whereby groups of macroeconomic variables, m_{t,i,G_1} and m_{t,i,G_2}, and forward rates, $f_t^{(n)}$, define a separate network. The collection of networks is then ensembled at the output layer.



(a) *Hybrid* network with fwd rates direct



(b) Macro network + fwd rate net



(c) Group-ensemble network

FIGURE 4: **Sample splitting strategy**

This figure shows the sample splitting used for cross-validation of the hyper-parameters of the penalized regressions, i.e., lasso, elastic net, ridge, and the neural networks. The forecasting exercise involves an expanding window that starts in January 1990. The full sample period is from 1971:08 to 2018:12. The blue area represents the sum of the training and validation sample. The former consists of the first 85% of the observations while the latter consists of the final 15% of observations. The training and the validation samples are consequential and not randomly selected in order to preserve the time series dependence. The red area represents the testing sample, which consists of the non-overlapping one-year holding period excess returns of treasury bonds.

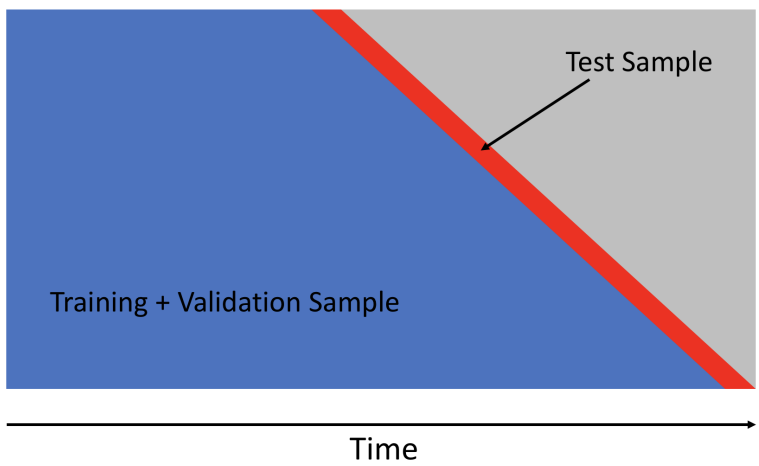
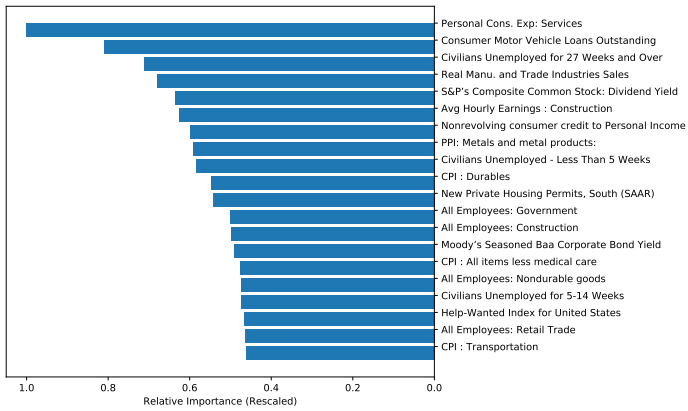
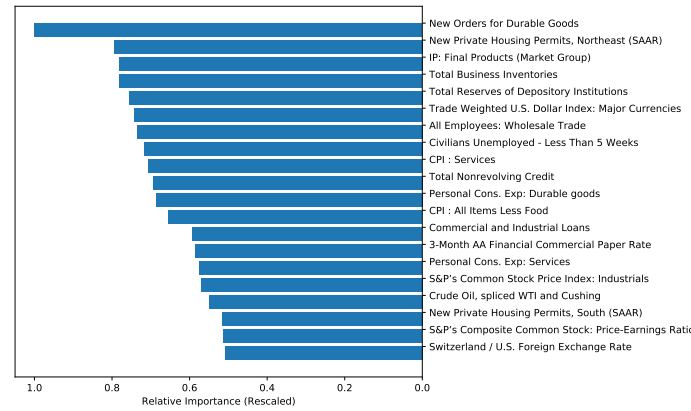


FIGURE 5: **Relative importance of macroeconomic variables**

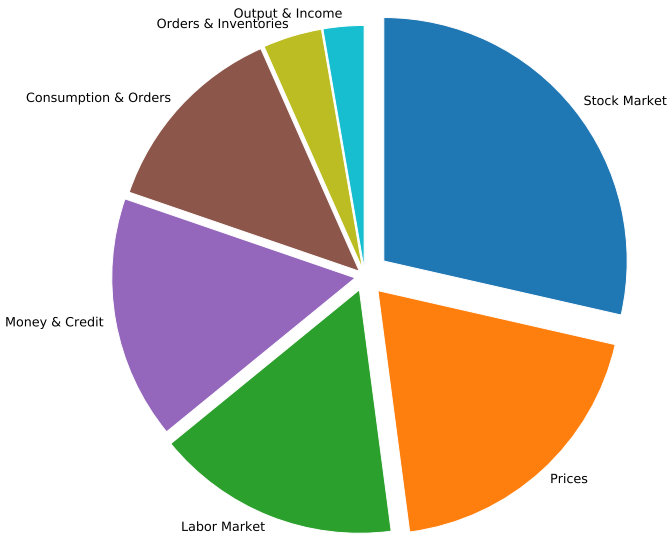
This figure shows the relative importance of different macro economic variables use to forecast bond excess returns. Panels (a) and (b) show results for individual variables, while panels (c) and (d) present results for groups of macro variables. The groups are labelled according to McCracken and Ng (2016). The relative importance of an input variable is computed based on the absolute value of the gradient of the network outputs with respect to the input variable. The gradient is evaluated at the in-sample mean of the input variable. The gradient-at-the-mean is calculated for each time t of the recursive forecasting exercise and then averaged over the out-of-sample period. For the grouped results in panels (c) and (d) the relative importance is averaged within groups.



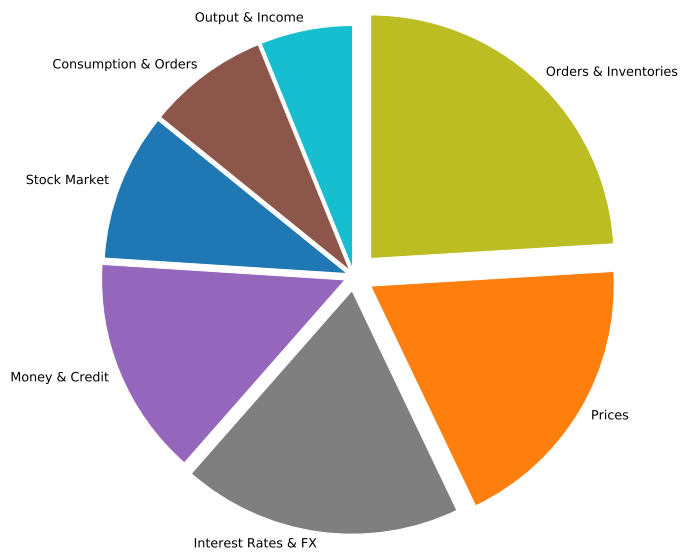
(a) 2-year maturity, individual variables



(b) 10-year maturity, individual variables



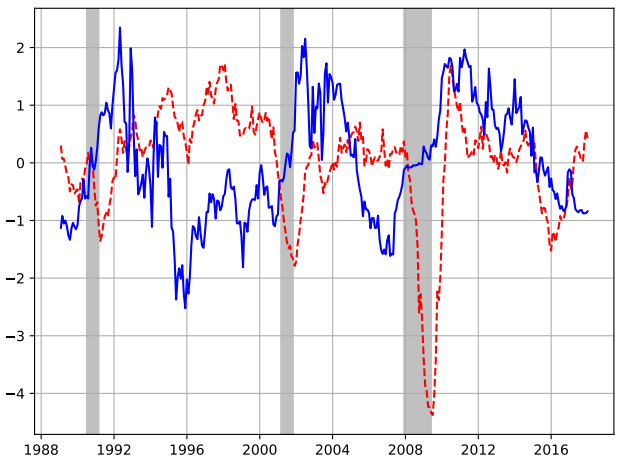
(c) 2-year maturity, groups of variables



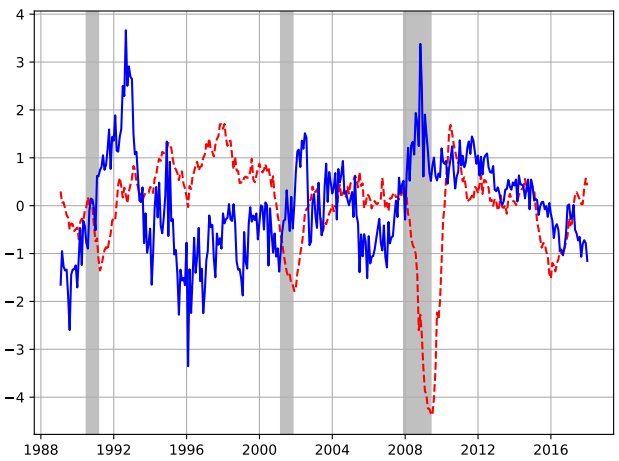
(d) 10-year maturity, groups of variables

FIGURE 6: **Bond excess returns, Model-implied risk premia, and Economic growth**

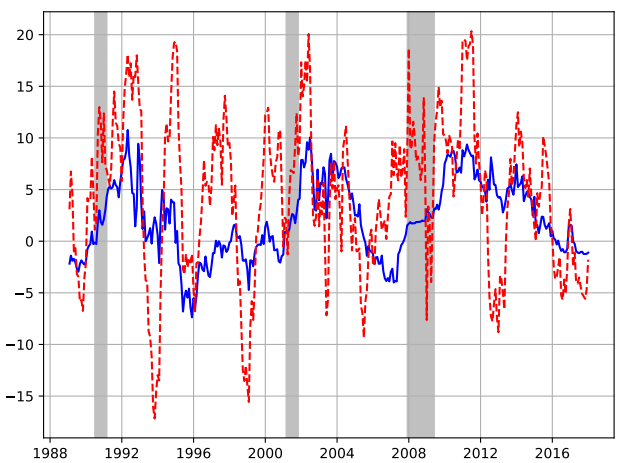
Panels (a) and (b) plot the model-implied expected bond excess returns for the 10-year maturity (blue lines) against the annual growth rate of industrial production in the US (red dashed line). Panels (c) and (d) display the time series of annual realized (red dashed line) and expected (blue line) ten-year bond excess returns (in percentage terms). We report the two best performing forecasts from the neural nets, that is the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables (see Table 1-2 for reference). The left panels report the results for the expected bond excess returns obtained by using the forward rates, whereas the right panels report the results for the expected bond excess returns obtained by using a large set of macroeconomic variables in addition to the forward rates.



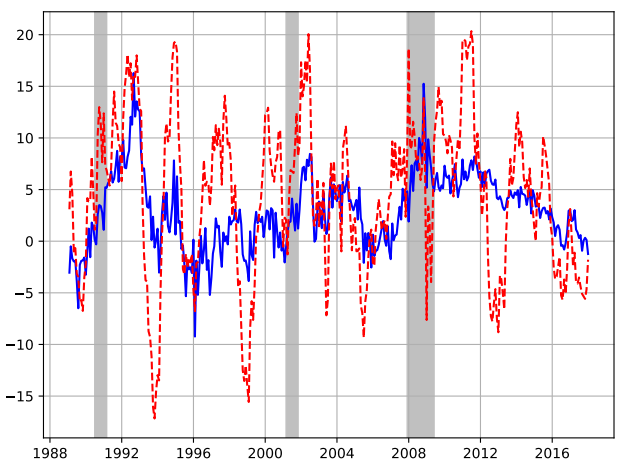
(a) Forwards only: Cyclical variation in expected returns



(b) Fwd rates + Macro: Cyclical variation in expected returns



(c) Forwards only: realized vs. expected bond returns



(d) Fwd rates + Macro: realized vs. expected bond returns

A A Simple Motivating Framework

Relying on the large literature on time varying risk premia (see e.g. Campbell and Cochrane, 1999, Wachter, 2006 and Buraschi and Jiltsov, 2007) we present a simple model with external habit formation which leads to the Quadratic Linear model.[1]

The representative agent maximizes

$$E\left[\int_0^\infty u(C_t, X_t, t)dt\right] ,$$

where the instantaneous utility function is given by

$$u(C_t, X_t, t) = \begin{cases} e^{-\rho t}\frac{(C_t - X_t)^{1-\gamma}}{1-\gamma} & \text{if } \gamma > 1 \\ e^{-\rho t}\log(C_t - X_t) & \text{if } \gamma = 1 \end{cases}$$

where X_t is an external habit level as in Campbell and Cochrane (1999). Consider now the Surplus Consumption Ratio

$$S_t = \frac{C_t - X_t}{C_t} .$$

We model external habit formation as in Pastor and Veronesi (2005), i.e. we assume:

$$\begin{aligned} S_t &= e^{s_t} \\ s_t &= a_0 + a_1 z_t + a_2 z_t^2 \\ dz_t &= k_z(\bar{z} - z_t)\,dt + \sigma_z dW_{c,t} . \end{aligned}$$

Pastor and Veronesi (2005) show that by choosing a_i appropriately (in particular, $a_2 < 0$), then $s_t < 0 \to S_t \in [0,1]$. In addition, we must have $\frac{\partial s(y)}{\partial y} = a_1 + 2a_2 y_t > 0$ so that positive shocks to consumption $dW_{c,t}$ translate into positive shocks to the surplus consumption ratio S_t. The rest of the model is defined by:

$$\begin{aligned} dc_t &= g_t dt + \sigma_c dW_{c,t} \\ dq_t &= i_t d_t + \sigma_q dW_{q,t} \end{aligned}$$

where we let $c_t = \log C_t$ and $q_t = \log Q_t$ be log consumption and log inflation. Finally assume that $\boldsymbol{X}_t = (g_t, i_t, z_t)'$ follows the process

$$d\boldsymbol{X}_t = K(\Theta - \boldsymbol{X}_t)dt + \Sigma d\boldsymbol{W}_t , \tag{A.1}$$

where $d\boldsymbol{W}_t$ is a vector of Brownian motions.
In this economy, the SDF is given by $M_t = e^{\eta t - \gamma(c_t + a_0 + a_1 z_t + a_2 y_t^2) - q_t}$ and the interest rate has a linear quadratic structure

$$r_t = \delta_0 + \gamma g_t + i_t + \delta_z z_t + \delta_{zz} z_t^2 \tag{A.2}$$

Finally, denote the zero coupon bond price by $P(\boldsymbol{X}_t, t; T)$. Given the specification of the model (A.1), the price of the zero coupon bond $P(\boldsymbol{X}_t, t; T)$ is the solution to the Partial Differential Equation (PDE)

$$rZ = \frac{\partial P}{\partial t} + \frac{\partial P}{\partial \boldsymbol{X}} K(\Theta - \boldsymbol{X}_t) + \frac{1}{2} tr\left(\frac{\partial^2 P}{\partial \boldsymbol{X} \partial \boldsymbol{X}'} \Sigma\Sigma'\right)$$

[1] This material is based on the 2015 Version of Pietro Veronesi's lecture notes on "Topics in Dynamic Asset Pricing".

subject to the final condition $P(\boldsymbol{X}_T, T; T) = 1$. Using the method of undetermined coefficients and exploiting the risk free rate equation (A.2), we can verify that the log bond price is given by

$$\log P(\boldsymbol{X}_t, t; T) = A(t; T) + \mathbf{B}(t, T)'\mathbf{X}_t + \mathbf{X}_t'\mathbf{C}(t; T)\mathbf{X}_t$$

where $A(t; T), \mathbf{B}(t, T)$ and $\mathbf{C}(t; T)$ satisfy a set of ODEs - see Ahn et al. (2002) and Leippold and Wu (2003).[2] Hence, the bond pricing formula is also linear-quadratic with factors given by consumption growth g_t, expected inflation i_t and habit z_t.

For a fairly general framework with non-linear dynamics under the historical measure, and encompassing many equilibrium models with recursive preferences and habit formation see Le et al. (2010). Finally, note that besides habit-based term structure models, non-linearities are also featured in state-dependent, learning-based models (see, e.g. Veronesi, 2004).

B Pairwise Test of Predictive Accuracy

We follow Gu et al. (2018) and implement a pairwise test as proposed by Diebold and Mariano (1995) (DM) to compare the predictions from different models. Diebold and Mariano (1995) show that the asymptotic normal distribution can be a very poor approximation of the test's finite-sample null distribution. In fact, the DM test can reject the null too often, depending on the sample size and the degree of serial correlation among the forecast errors. To address this issue, we adjust the DM test by making a bias correction to the test statistic as proposed by Harvey et al. (1997).

Figure B.1 confirms the statistical significance of the conclusions reached in Section 4.1 and 4.2 by implementing such test. The figure reports the significance of the performance gaps while the direction can be inferred by looking at Tables 1-2. We color-coded the statistical significance of the test, with lighter color suggesting a stronger rejection of the null (H_0: the pairs have the same performance).

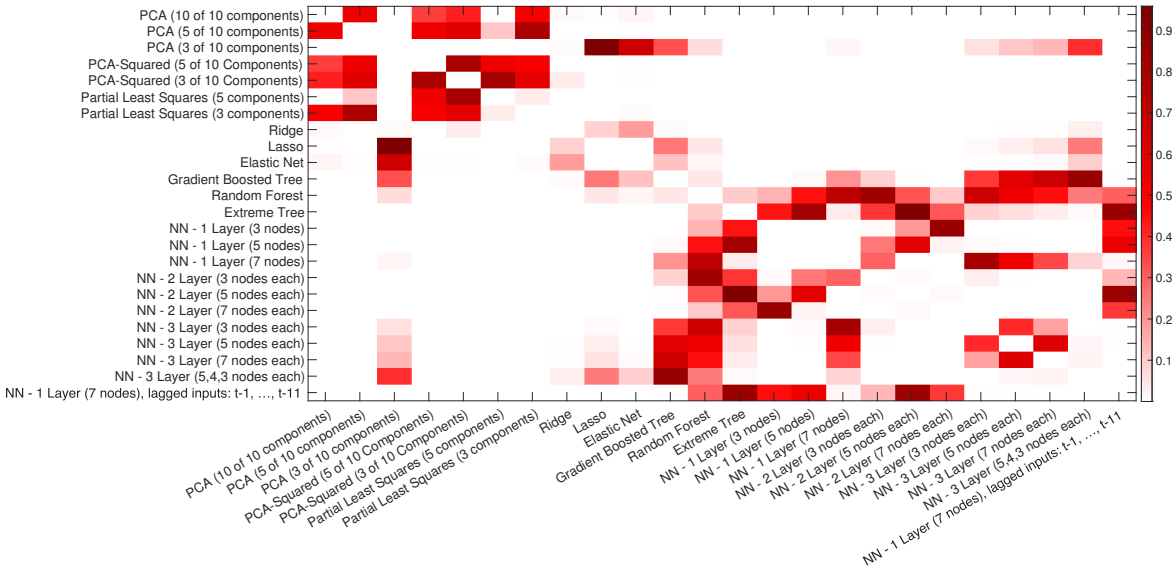
The top panel reports the analysis when yields are the only predictors. We observe that: (1) the performances across classes of machine learning methods tend to differ markedly, with penalized regressions that have significantly worse accuracy relative to trees, and NNs (white cells); (2) shallow learners and NNs with 2-layers often tend to have similar performance (darker red); however, increasing the depth of the NN further worsen the performance (lighter red); (3) lagged forward rates do not improve the predictive accuracy of a shallow learner (red cell).

The bottom panel reports the pairwise DM tests when the macro variables from the FRED-MD database are added to forward rates. The test confirms that we observe that the depth of the network matters, as testified by the statistically significant out-performance (lighter cells) of a three-layer hybrid network (NN 3 Layer, fwd rates direct) over a shallow and two-layer hybrid network. At the same time the figure confirms that a shallow network with group ensembling performs on par with a deeper, three-layer network that does not distinguish between economic categories.

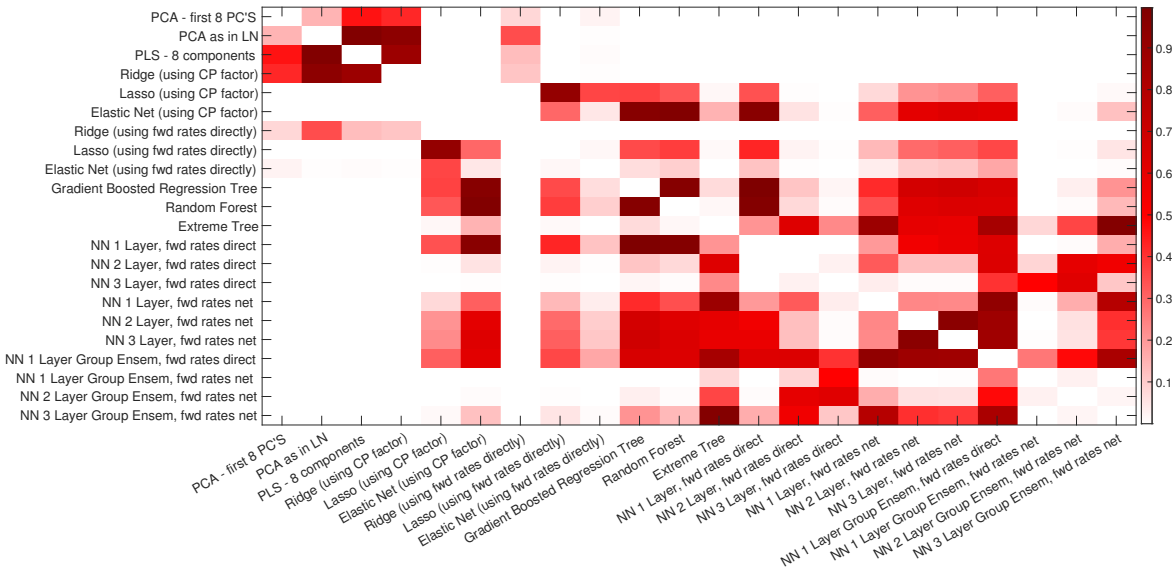
[2] More precisely, we conjecture $P(\boldsymbol{X}_t, t; T) = e^{A(t;T)+\mathbf{B}(t,T)'\mathbf{X}_t+\mathbf{X}_t'\mathbf{C}(t;T)\mathbf{X}_t}$, we compute derivatives $\frac{\partial P}{\partial t}$, $\frac{\partial P}{\partial \boldsymbol{X}}$, and $\frac{\partial^2 P}{\partial \boldsymbol{X}\partial \boldsymbol{X}'}$, we substitute r and partial derivatives in the PDE, and we collect terms.

FIGURE B.1: **Pairwise tests of predictive accuracy**

This figure shows the results of a pairwise test of predictive accuracy based on Diebold and Mariano (1995) and extended by Harvey et al. (1997). The figure reports the results when forecasts are based only on the forward rates (top panel) and when both macroeconomic variables and forward rates are used (bottom panel). The red color scale indicates the statistical significance of the test, that is the lighter the color the lower the p-value for the null hypothesis that the forecasts between a pair are not statistically different.



(a) Forecast with forward rates



(b) Forecast with macro + fwd rates

C Dissecting predictability: Further discussion and additional results

C.1 Bond Return Predictability and Cumulative SSE

To identify the periods in which the models perform best, we follow Welch and Goyal (2008) and compute the difference in the cumulative sum of squared errors (SSE) for the EH model versus the machine learning model of interest, $\Delta CumSSE$. Positive and increasing values of $\Delta CumSSE$ suggest that the model under consideration generates more accurate point forecasts than the EH benchmark.

Figure C.1 plot the $\Delta CumSSE$ for the best performing regression tree specification, i.e., extreme tree (Panels (a) and (c)), and for the best performing neural network, namely the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates (Panel (b)) – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables (Panel (d)) – (see Table 1-2 for reference). We focus on the ten-year bond maturity.

[Insert Figure C.1 about here]

In general the plots show that the various machine learning models perform well relative to the EH model as manifested by lines that are increasing for most periods. Indeed, only when we consider exclusively yields-based predictors, we then observe occurrence of decreasing graphs (i.e. periods where the models underperform against this benchmark). However, these are few isolated occasions (e.g., around 2001 and 2008). Interestingly, adding macroeconomic variables improves the predictive accuracy of the models relative to the benchmark. Comparing panel (c) to (d), we note that: (1) the performance of extreme trees is particularly effective in few, rather prolonged, periods, namely 1992 to 1996 and the aftermath of the financial crisis; (2) the performance of the NN with group ensembling is instead characterized by an almost steady improvement in performance.

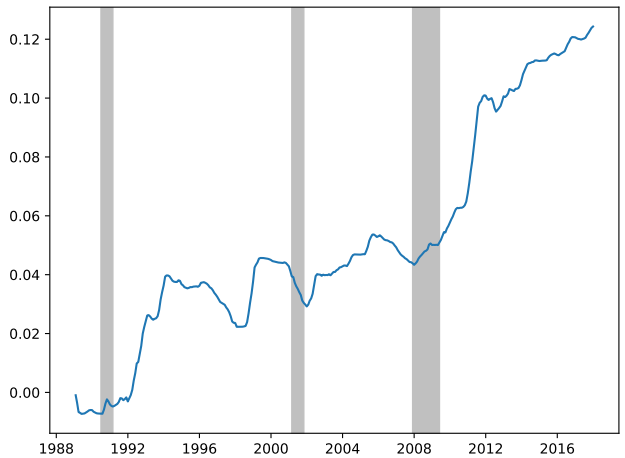
Another interesting aspect to investigate is whether expected bond returns could have evolved differently in response to availability of technology.[3] However, referring to our plots of out-of-sample R^2 over time, even in later years when adoption of neural networks spread, we still notice the outperformance of neural networks vis-a-vis the expectation hypothesis.

Overall, we take these patterns as rather reassuring that the value-added through machine learning based forecasts is rather pervasive and not concentrated in isolated events.

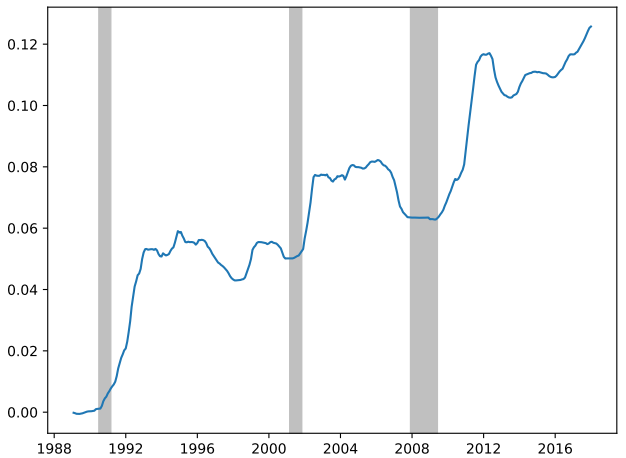
[3]The theory necessary to apply our networks was available in the 1990 when our out-of-sample period starts. The overarching concept of back-propagation in multi-layer neural networks was introduced in Werbos (1974). The use of automatic differentiation (AD) that is necessary for fast computation of gradients during back-propagation was used in Werbos (1982). An early example of a study using neural networks for prediction in a finance context, on gas markets to be precise, with similar models as used by us is Werbos (1988) and others exist before 1990. What is less clear is whether the necessary computational power was available to estimate the models in a reasonable amount of time. Also, while the most fundamental building blocks were in place in the late 1980s, regularization concepts such as dropout (Srivastava et al., 2014) and batch normalization (Ioffe and Szegedy, 2015) were only introduced later.

FIGURE C.1: **Squared forecast errors across time**

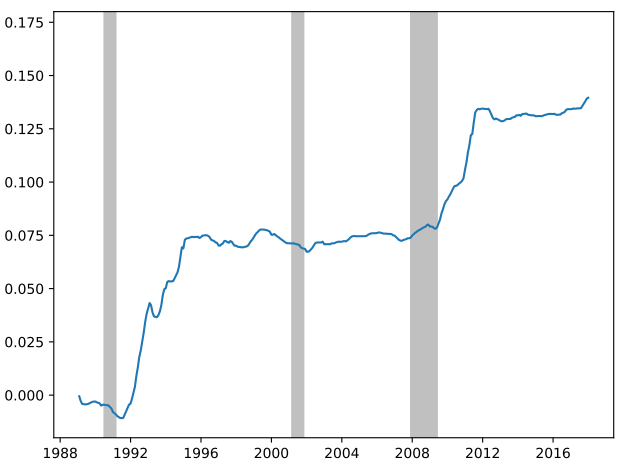
This figure plots the time series of difference in squared forecast errors from a given model versus the expectations hypothesis. We scale the forecast errors by the variance of the dependent variable times $T-1$, i.e. each month t we plot the value attained by $\frac{(\hat{\epsilon}_{t+1}^{EH})^2-(\hat{\epsilon}_{t+1}^{Model})}{(T-1)\text{Var}(rx_{t+1})}$. The out-of-sample period starts in 1990. The expectation hypothesis uses all data starting from the in-sample period in 1971:08. The figures present results for the 10-year maturity and focus on the best performing regression tree and neural network when forecasts are generated either based on forward rates only or by macroeconomic variables + forward rates.



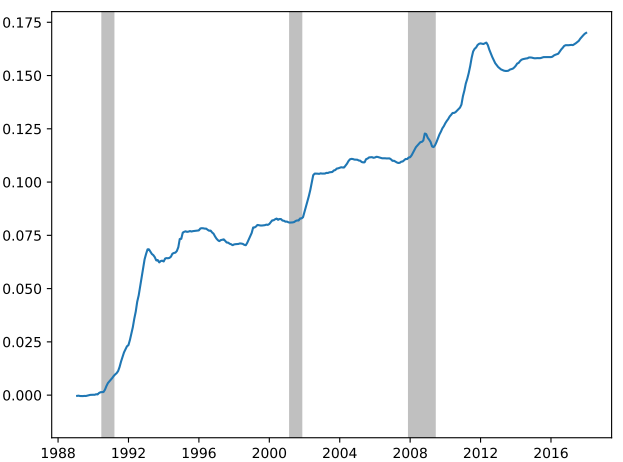
(a) Extreme tree, forward rates, 10-year maturity



(b) Neural net, forward rates, 10-year maturity



(c) Extreme tree, macro + fwd rates, 10-year maturity



(d) Neural net, macro + fwd rates, 10-year maturity

C.2 Bond Return Predictability and Shape of the Yield Curve

TABLE C.1: **Conditional Forecast Accuracy: Double Sorts on Prevailing Yield Curve Level & Slope**

This table reports the forecast accuracy when conditioning on the prevailing shape of the yield curve, i.e. its level and slope. As a measure of of yield curve level we use the 2-year yield, while the measure of yield curve slope is the difference between the 10-year and 2-year yield. We sort all observations in our out-of-sample period based on the median level and slope of the yield curve prevailing at the start of the holding period. The double sort is performed unconditionally. We denote observations below the median by "Low" and observations above the median by "High". For example, "Level Low - Slope High" refers to all observations for which the prevailing yield curve level was below the median, while the yield curve slope was above the median. The median yield curve level over our out-of-sample period is 3.84% and the median yield curve slope is 1.33%. Forecast accuracy is proxied by the R^2 in regressions of realized returns, $xr_{t+1}^{(i)}$, on the predicted bond risk premium, $\hat{xr}_{t+1}^{(i)}$: $xr_{t+1}^{(i)} = \alpha + \beta \hat{xr}_{t+1}^{(i)} + \epsilon_{t+1}^{(i)}$, where the regressions are performed using all the observations that fall into the four distinguished yield curve shape cases. Predicted bond risk premia stem from forecasting either with only the forward rates (Panel A) – using *NN 1 Layer (3 nodes)* – or with forward rates plus macroeconomic variables (Panel B) – using *NN 1 Layer Group Ensem + fwd rate net* – (see Table 1-2 for reference). The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12. For each case, we report R^2, the mean fitted value, the p-values for the hypothesis tests $H_0: \alpha = 0$ and $H_0: \beta = 1$, and the fraction of the sample period falling into the respective case.

Panel A: Forecasting with Forward Rates

	R^2 (%)	Mean Fitted Value	p-val ($\alpha = 0$)	p-val ($\beta = 1$)	Sample Fraction
All	21.97	4.33%	0.69	0.10	100.0%
Level Low - Slope Low	7.79	-2.27%	0.00	0.91	9.8%
Level Low - Slope High	16.18	6.26%	0.76	0.70	40.2%
Level High - Slope Low	23.79	3.60%	0.00	0.15	40.2%
Level High - Slope High	58.84	5.96%	0.09	0.00	9.8%

Panel B: Forecasting with Forward Rates + Macro

	R^2 (%)	Mean Fitted Value	p-val ($\alpha = 0$)	p-val ($\beta = 1$)	Sample Fraction
All	27.95	4.33%	0.91	0.21	100.0%
Level Low - Slope Low	14.06	-2.27%	0.00	0.99	9.8%
Level Low - Slope High	18.48	6.26%	0.83	0.42	40.2%
Level High - Slope Low	28.16	3.60%	0.01	0.12	40.2%
Level High - Slope High	65.75	5.96%	0.06	0.01	9.8%

C.3 Relative Importance of Macroeconomic Variables: Further Discussion

To study the relative importance of macroeconomic variables we rely on the partial derivative of the target variable with respect to sample average of each input. These partial derivatives represent the sensitivity of the output to the ith input, conditional on the network structure and the average value of the other input variables (see Dimopoulos et al., 1995), and are akin to the betas of a simple linear regression.

In principle, the partial derivative of the target variable with respect to the sample average of each input, see equation (7) is a deterministic object. As such, one may question the robustness and reliability of our estimates given the stochastic nature of the training process of neural networks (see, e.g., Hansen and Salamon, 1990 and Dietterich, 2000). The training process of NNs is stochastic in so far that the optimization path will generally depend on its initialization. Therefore, different initializations of the network weights can lead to different outcomes. To address this issue, we estimate the relative importance from different starting values of the neural network weights, and average the results. To be precise, we initialize network weights using the He (2015a) normal initialization procedure and use 100 different, but fixed seeds. Then we construct predictions from all network estimates and calculate the corresponding partial derivatives as in equation (7). The final result is obtained by averaging across the estimated models.

Alternative methodologies for measuring relative variable importance in NNs have been proposed in the literature. For instance, Sung (1998) proposed a stepwise method that consists of adding or rejecting one input variable at a time and noting the effect on the Mean Squared Error (MSE). Based on the changes in MSE, the input variables can be ranked according to their importance in several different ways depending on different arguments. The major drawback of stepwise methods is that at each step of the selection algorithm a new model is generated that requires training. For an exercise like ours, in which there are more than 120 input variables and forecasts are generated recursively this is computationally too expensive to execute. Alternatively, Lek et al. (1995) and Lek et al. (1996) propose to study each input variable successively when the others are blocked at fixed values. Depending on the users' needs one can set single inputs to their sample mean, their median, or simply to zero (see, e.g., Gu et al., 2018). The relative importance of a given input is then investigated by looking at the changes in the MSE.

Figure 5 shows the relative importance of each input variable based on the gradient in equation (7). The figure shows the rescaled value of the gradient such that a value of zero means that a variable is least relevant and a value of one means that a variable is the most important in relative terms. The gradient-at-the-mean is calculated for each time t of the recursive forecasting, then averaged over the out-of-sample period.

C.4 Interactions Within or Across Categories: Results

TABLE C.2: **Magnitude of Cross- and Within-group Interactions**

This table reports the sum of the absolute value of the cross-group (Panel A) and within-group (Panel B) interactions obtained from an ensembled neural network with one hidden layer ("Groups ensemble") and a (non-ensembled) neural network with three hidden layers ("Fully connected network"). See Table 2 for their performance (row "NN 3 Layer (32, 16, 8 nodes), fwd rates direct" and "NN 1 Layer Group Ensem (1 node per group), fwd rates Net (1 layer: 3 nodes)"). Interactions magnitudes are computed by numerically approximating the network Hessian's, i.e. the second derivatives of network outputs with respect to two distinct network inputs. The NNs take as inputs both macroeconomic variables and forward rates. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Panel A: Sum of Cross-Group Hessian Absolute Values

Model	2y	3y	4y	5y	7y	10y
Fully connected network	60.24	108.61	153.22	184.43	247.84	336.49
Groups ensemble	0.04	0.07	0.09	0.11	0.14	0.18

Panel B: Sum of Within-Group Hessian Absolute Values

Model	2y	3y	4y	5y	7y	10y
Fully connected network	11.93	21.30	30.12	36.03	48.51	65.77
Groups ensemble	16.93	32.05	45.09	54.70	78.85	117.05

C.5 Model Uncertainty: Results

TABLE C.3: **Alternative Model Combination Strategies**

This table reports the out-of-sample R^2_{oos} obtained using a large panel of macroeconomic variables and forward rates to predict annual excess bond returns for different maturities. In addition to the best performing neural network – *NN 1 Layer Group Ensem + fwd rate net* (see Table 2), we compute the R^2_{oos} for three alternative model combination strategies. The first is a recursive full cross-validation that selects every five years not only the dropout rate and the L1/L2 penalties, but also the number of layers, the nodes per macro group, and the nodes in fwd rate net. The second and third model combination are based on a weighted average of each neural network forecasts with weights that are calculated as the inverse of the validation loss or, alternatively, simply as $1/N$ where N is the number of neural networks estimated. The out-of-sample R^2_{oos} is calculated by considering the expectations hypothesis as a benchmark. That is, we compare the forecasts obtained from each methodology to the prediction based on the historical mean of bond excess returns. In addition to the R^2_{oos} we report the p-value calculated as in Clark and West (2007) in parentheses. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Models	R^2_{oos}						R^2_{oos} EW
	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(EW)}_{t+1}$
NN 1 Layer Group Ensem (1 node per group), fwd rate net (1 layer: 3 nodes)	20.0%	25.6%	29.5%	31.2%	33.6%	36.3%	34.0%
	(0.002)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Inverse Val. Loss Weighted Model (across all NNs)	22.3%	25.6%	29.1%	30.8%	32.0%	34.4%	32.6%
	(0.002)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Equally Weighted Model (across all NNs)	22.0%	25.2%	28.6%	30.4%	31.6%	33.9%	32.1%
	(0.002)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
Full CV Model	24.7%	27.6%	30.4%	31.7%	32.3%	34.7%	33.4%
	(0.001)	(0.001)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)

TABLE C.4: **Stability of the Neural Network Ranking**

This table reports how often the four best performing neural networks are selected throughout the sample. More specifically, we select the top 4 models based on the unconditional average of (inverse) validation loss. Then we count how often the four models rank 1st, 2nd, 3rd and 4th, in terms of their inverse validation error. The sum of the values in each column equals the number of periods in our out-of-sample period. This gives an approximate measure of how often one model ranks on top in terms of validation loss. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

	Ranking			
Model	1st place	2nd place	3rd place	4th place
NN 1-Layer Group Ensem + Fwd. Rate Net	170	42	115	21
NN 1-Layer Group Ensem + Fwd. Rates	137	50	154	7
NN 3-Layer + Fwd. Rates	31	112	16	189
NN 2-Layer Group Ensem + Fwd. Rates Net	10	144	63	131

C.6 Model Performance and Lagged Macroeconomic Variables

We assess the robustness of the performance of our non-linear methods with respect to the timing of the macroeconomic predictors. Information flow to investors could occur with a lag such that macroeconomic information is not available right away, therefore leading to our models overfitting. In order to rule out that this is the case, we follow Rapach and Zhou (2019) and lag all macroeconomic variables by one month to account for data becoming available with a delay. Note, we do not lag market-based variables including interest rates, exchange rates, stock market data and the oil price since these variables are available to investors virtually in real-time.

In Table C.5 we summarize the results of the recursive forecasting exercise with lagged macroeconomic variables. Overall, we find that the performance of the non-linear models that are at the center of our analysis is robust to lagging the macroeconomic predictors by one month. Also, the ordering of models in terms of their predictive performance is basically unchanged: the set of neural networks outperforms the regression tree approaches as long as the NNs are deep enough or use group ensembling. We continue to find that a shallow group ensembled neural network performs on par with or better than a deeper neural network that does not group variables according to their macroeconomic character. In particular, the ensembled neural network with 1 hidden layer per group of macroeconomic variables and a network for the forward rates continues to perform best out of all models studied: the overall model performance (R^2 EW of 34.0%, c.f. Table 2) without lagging the predictors is on par with the model performance when predictors are lagged (R^2 EW of 32.8%).

Finally, let us observe that Table 2 and Table C.5 rely on two different frameworks, and demand to be interpreted accordingly. Table 2 relies on the "economic agents know" framework: the econometrician has limited information relative to *economic agents who*, in contrast, *know* the history of prior macroeconomic data and how bond returns react to real quantities. This is the relevant framework for us, since our main objective is, through the lens of machine learning methods, to provide a better understanding of the dynamics of bond risk premia and its economic drivers, rather than to create a trading strategy. For further discussions on how to interpret out-of-sample tests see, e.g., Atanasov, Moller and Priestley (2019).

TABLE C.5: **Forecasting Annual Holding-period Returns with Forward Rates and Lagged Macroeconomic Variables**

This table reports the out-of-sample R^2_{oos} obtained using forward rates and a large panel of macroeconomic variables to predict annual excess bond returns for different maturities when macroeconomic variables are lagged by one month. Given that macroeconomic variables that are market-based are available in close to real-time, variables pertaining to the interest rates, exchange rates, stock market categories as well as oil price are not lagged. All other variables are lagged by one month. To compute the out-of-sample R^2_{oos} we compare the forecasts obtained from each methodology to the expectation hypothesis (i.e., prediction based on the historical mean). In addition to the R^2_{oos} we report the p-value for the null hypothesis $R^2_{oos} \leq 0$ calculated as in Clark and West (2007). Notice that we report a p-value only when the R^2_{oos} is positive. The out-of-sample prediction errors are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12. Penalized regressions are estimated including macro-economic variables plus either raw forward rates or a linear combination of forward rates as introduced by Cochrane and Piazzesi (2005) (CP). Similarly, neural networks are estimated either adding the CP factor as an additional regressor in the output layer ("fwd rates direct") or by estimating a separate network for forward rates and ensembling both macro and forward rates networks in the output layer ("fwd rates net").

Models	R^2_{OOS}						p-value						R^2_{OOS} EW	p-value EW
	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(2)}_{t+1}$	$xr^{(3)}_{t+1}$	$xr^{(4)}_{t+1}$	$xr^{(5)}_{t+1}$	$xr^{(7)}_{t+1}$	$xr^{(10)}_{t+1}$	$xr^{(EW)}_{t+1}$	$xr^{(EW)}_{t+1}$
Gradient Boosted Regression Tree	15.3%	18.3%	15.5%	22.2%	24.2%	21.7%	0.000	0.003	0.006	0.003	0.004	0.005	24.8%	0.3%
Random Forest	20.1%	13.5%	15.5%	20.4%	25.6%	29.8%	0.007	0.006	0.006	0.002	0.002	0.001	27.7%	0.1%
Extreme Tree	22.8%	14.1%	16.5%	23.0%	23.4%	30.4%	0.005	0.004	0.006	0.002	0.004	0.002	26.2%	0.3%
NN 1 Layer (32 nodes), fwd rates direct	9.3%	11.3%	17.5%	19.9%	22.9%	26.3%	0.002	0.001	0.000	0.000	0.000	0.000	21.8%	0.0%
NN 2 Layer (32, 16 nodes), fwd rates direct	16.5%	21.5%	24.7%	27.5%	29.0%	31.6%	0.000	0.000	0.000	0.000	0.000	0.000	28.6%	0.0%
NN 3 Layer (32, 16, 8 nodes), fwd rates direct	22.2%	25.6%	29.8%	31.5%	32.1%	34.4%	0.000	0.000	0.000	0.000	0.000	0.000	32.6%	0.0%
NN 1 Layer (32 nodes), fwd rates net (1 layer: 3 nodes)	4.2%	15.0%	19.8%	24.7%	26.5%	29.6%	0.006	0.003	0.002	0.001	0.001	0.001	25.9%	0.1%
NN 2 Layer (32,16, nodes), fwd rates net (1 layer: 3 nodes)	12.6%	19.7%	23.2%	25.5%	27.1%	29.2%	0.006	0.002	0.001	0.001	0.001	0.001	27.1%	0.1%
NN 3 Layer (32,16, 8 nodes), fwd rates net (1 layer: 3 nodes)	10.0%	17.0%	22.1%	25.3%	27.5%	30.9%	0.014	0.005	0.003	0.001	0.001	0.001	27.2%	0.1%
NN 1 Layer Group Ensem (1 node per group), fwd rates direct	22.1%	25.6%	28.5%	30.7%	30.8%	32.0%	0.001	0.000	0.000	0.000	0.000	0.000	31.3%	0.0%
NN 1 Layer Group Ensem (1 node per group), fwd rate net (1 layer: 3 nodes)	19.0%	24.5%	28.7%	30.8%	32.8%	34.9%	0.003	0.002	0.001	0.001	0.001	0.001	32.8%	0.1%
NN 2 Layer Group Ensem (2,1 nodes per group / hidden layer), fwd rate net (2 layer: 3 nodes)	14.8%	21.7%	25.6%	28.0%	29.5%	31.5%	0.008	0.003	0.002	0.001	0.001	0.000	29.9%	0.1%
NN 3 Layer Group Ensem (3, 2, 1 nodes per group / hidden layer), fwd rate net (3 layer: 3 nodes)	13.7%	20.5%	24.8%	27.1%	29.1%	32.0%	0.011	0.004	0.002	0.001	0.001	0.001	29.1%	0.1%

D Economic Value of Excess Bond Returns Forecasts: Additional Discussions and Tests

D.1 The Asset Allocation Framework for a Power Utility Investor

We consider the investment decision of an agent that selects the weights on the risky n-period bonds $\boldsymbol{w}_t = \left[w_t^{(2)} \dots w_t^{(10)}\right]$ versus a one-period bond that pays a risk-free rate equal to $y_t^{(1)}$. In the main text we discuss an investor with mean-variance utility; here we discuss an extended framework whereby a representative investor has a power utility of the form,

$$U\left(\boldsymbol{w}_t, \boldsymbol{xr}_{t+1}\right) = \frac{\left[\left(1 - \boldsymbol{w}_t'\iota\right)\exp\left(y_t^{(1)}\right) + \boldsymbol{w}_t'\exp\left(y_t^{(1)}\iota + \boldsymbol{xr}_{t+1}\right)\right]^{1-\gamma}}{1-\gamma}, \qquad \gamma > 0 \tag{D.1}$$

where γ captures the investor's risk aversion and ι is a vector of ones. We follow Campbell and Viceira (1999), Campbell and Viceira (2004) and Gargano et al. (2019) and assume excess bond returns are jointly log-normal distributed so that the excess returns on a portfolio of treasury bond can be approximated by

$$R_{p,t+1} = 1 + y_t^{(1)} + \boldsymbol{w}_t'\boldsymbol{xr}_{t+1} + \frac{1}{2}\boldsymbol{w}_t'\boldsymbol{\sigma}_{t+1|t}^2 - \frac{1}{2}\boldsymbol{w}_t'\Sigma_{t+1|t}\boldsymbol{w}_t \tag{D.2}$$

where $\Sigma_{t+1|t}$ denotes the covariance matrix of the excess bond returns, and we denote with $\boldsymbol{\sigma}_{t+1|t}$ its diagonal elements. Campbell and Viceira (2004) show that under log-normality of excess returns, the optimal allocation on a maturity-specific bond can be defined as

$$w_t^{(n)} = \frac{1}{\gamma\left(\sigma_{t+1|t}^{(n)}\right)^2}\left[\widehat{xr}_{t+1}^{(n)} + \left(\sigma_{t+1|t}^{(n)}\right)^2/2\right] \tag{D.3}$$

Given these optimal set of weights, the realized utility for, say, the univariate case can be computed by plugging $\boldsymbol{w}_t$ into equation (D.1).

Similar to the mean-variance case (see Section 6.1) we proxy for $\Sigma_{t+1|t}$ by using a rolling sample variance estimator as in Thornton and Valente (2012), and set the coefficient of relative risk aversion equal to $\gamma = 5$. Further, to test if the annualized CER values are statistically greater than zero, we employ again a Diebold and Mariano (1995) test. Specifically, for a power utility investor that selects a single risky asset with maturity n using when the forecast from a NN, we estimate the regression

$$u_{t+1,NN}^{(n)} - u_{t+1,EH}^{(n)} = \alpha^{(n)} + \varepsilon_{t+1}$$

where

$$u_{t+1,j}^{(n)} = \frac{1}{1-\gamma}\left[\left(1 - \omega_{t,j}^{(n)}\right)\exp(y_t^{(1)}) + \omega_{t,j}^{(n)}\exp(y_t^{(1)} + xr_{t+1}^{(n)})\right]^{1-\gamma}$$

and $j = \{EH, NN\}$.

D.2 Further Results on the Economic Value of Return Forecasts

In addition to the benchmark case that restricts the weights to be in the interval $-1 \leq w_t^{(n)} \leq 2$ to prevent extreme investments, we also consider two alternative scenarios. The first scenario restricts the optimal weights to be non-negative, i.e., $w_t^{(n)} \in (0, 0.99)$. Such scenario ensures that the expected utility is finite even in the case of unbounded returns (see Geweke, 2001b for a discussion). The second scenario leaves the portfolio weights unrestricted but instead restricts the bond returns to fall in the interval $(-100\%, +100\%)$. Similar to restricting the weights to be non-negative, such restriction prevents the expected utility from being unbounded (see Johannes et al., 2014).

The results in Table D.1 confirm our results in the main text (c.f. Section 6). For instance, Panel A shows that including macroeconomic information tend to improves the economic performance with respect to a model which exploits only information from the yield curve, and that forecasts from the best performing neural network achieves a higher utility with respect to a competing non-linear model such as extreme trees.

Panel B shows that by winsorizing the returns instead of the optimal weights the conclusion remains qualitatively the same; again macroeconomic information brings additional economic value and even more so when neural networks are used. As a whole, these results are in line with the main evidence described in Section 6.

TABLE D.1: **Economic Significance of Bond Predictability: Robustness**

This table reports two robustness exercises concerning the out-of-sample economic performance reported in Table 5. Panel A reports the results for the weights restricted to be non-negative, whereas Panel B restricts the realized returns on each maturity to be between -100% and 100%. We report results for an investor with either mean-variance or power utility and a coefficient of risk aversion equal to five. The models are benchmarked against the expectation hypothesis. The table reports both multi-asset results and the case of a single risky asset. In the univariate asset allocation case, the investor selects either the two- or the ten-year bond, along with the one-year short-rate. In the multivariate case, the investor selects bond excess returns across the six maturities, two- to five-, seven- and ten-years. The asset allocation decision is based on the predictions implied by either the best performing regression tree specification, i.e., extreme tree, or the best performing neural network, namely the *NN 1 Layer (3 nodes)* – when forecasting with only the forward rates – and *NN 1 Layer Group Ensem + fwd rate net* – when including also macroeconomic variables – (see Table 1-2 for reference). The row Δ reports the value added by NN relative to extreme tree within each application ("Fwd rates" and "Fwd + Macro"). The column Δ reports the value added by "Fwd+Macro" relative to "Fwd rates" within each model (NN and extreme tree). The models are benchmarked against the expectation hypothesis. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12. Statistical significance is based on a one-sided Diebold and Mariano (1995) test extended by Harvey et al. (1997) to account for autocorrelation in the forecasting errors. We flag in bold those values that are statistically significant at the 5% confidence level.

Panel A: Non-negative weights

		2-year maturity			10-year maturity			All		
		Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ
Mean-Variance	Neural net	0.017	**0.025**	0.007	**0.915**	**1.242**	**0.318**	**2.340**	**3.867**	**1.527**
	p-value	(0.322)	(0.000)	(0.557)	(0.011)	(0.000)	(0.041)	(0.010)	(0.011)	(0.000)
	Extreme tree	0.034	0.011	-0.023	**0.958**	**0.863**	0.159	**2.812**	**3.376**	**0.555**
	p-value	(0.416)	(0.512)	(0.449)	(0.003)	(0.000)	(0.473)	(0.009)	(0.009)	(0.004)
	Δ	-0.017	0.013		**-0.043**	**0.379**		**-0.472**	**0.491**	
	p-value	(0.597)	(0.471)		(0.261)	(0.014)		(0.023)	(0.015)	
Power Utility	Neural net	**0.045**	**0.098**	**0.053**	**1.235**	**1.765**	**0.530**	**2.624**	**4.133**	**1.509**
	p-value	(0.000)	(0.000)	(0.010)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
	Extreme tree	0.007	**0.075**	**0.068**	**1.507**	**1.198**	**-0.308**	**2.568**	**2.995**	0.427
	p-value	(0.763)	(0.000)	(0.004)	(0.000)	(0.000)	(0.010)	(0.000)	(0.000)	(0.242)
	Δ	**0.038**	0.023		-0.272	**0.566**		0.056	**1.138**	
	p-value	(0.034)	(0.057)		(0.055)	(0.000)		(0.820)	(0.000)	

Panel B: Winsorized returns

		2-year maturity			10-year maturity			All		
		Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ	Fwd rates	Fwd + Macro	Δ
Mean-Variance	Neural net	0.044	0.088	0.045	**0.910**	**2.331**	**1.413**	**2.551**	**3.143**	**0.592**
	p-value	(0.110)	(0.091)	(0.098)	(0.000)	(0.000)	(0.000)	(0.011)	(0.006)	(0.012)
	Extreme tree	0.028	0.078	0.050	**1.191**	**2.171**	**0.970**	1.156	**2.756**	**1.600**
	p-value	(0.261)	(0.076)	(0.094)	(0.023)	(0.000)	(0.000)	(0.063)	(0.001)	(0.002)
	Δ	0.016	0.009		-0.282	0.160		**1.395**	**0.387**	
	p-value	(0.171)	(0.519)		(0.067)	(0.087)		(0.016)	(0.023)	
Power Utility	Neural net	**0.056**	**0.110**	**0.054**	**1.170**	**3.143**	**1.961**	**2.294**	**4.411**	**2.117**
	p-value	(0.012)	(0.011)	(0.034)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
	Extreme tree	0.022	0.023	0.001	**1.518**	**2.681**	**1.162**	**2.601**	**3.712**	**1.111**
	p-value	(0.486)	(0.324)	(0.577)	(0.000)	(0.000)	(0.006)	(0.000)	(0.000)	(0.024)
	Δ	0.034	**0.087**		**-0.348**	**0.459**		-0.307	**0.697**	
	p-value	(0.093)	(0.032)		(0.041)	(0.034)		(0.169)	(0.039)	

D.3 Economic Drivers of Portfolio Performance

In this section, we investigate the potential drivers of the economic gains from bond return predictability. Intuitively one may expect large economic gains during bad times, when uncertainty and disagreement are large. Indeed Sarno et al. (2016) find large economic gains from predictability of bond returns during times with high macroeconomic uncertainty.

Specifically, we study the relation between the realized utility obtained in the portfolio analysis of Section 6 and the structural risk factors presented in Section 7.2. Table D.2 reports the results when we employ as predictors the macro variables from the FRED-MD database in addition to forward rates. Panels A and B show the analysis for the CER obtained by a mean-variance and power utility investor, respectively.

Most of the results confirm the evidence in Table 7 for our forecasts of excess bond returns. Indeed, across panels, we find that bond volatility is only weakly related to realized utility. The link between our realized utility and nominal disagreement is positive and statistically significant whereas we find only a weak association with real disagreement. This confirms the recent findings of Gargano et al. (2019) that inflation disagreement is an important driver of portfolio performance.[4] Other variables stand out as important determinants of variation in realized utility, namely the risk aversion proxies and (macro and inflation) uncertainty. Interestingly, in a kitchen sink regression where we preselect predictors based on their statistical significance (specification (vi)), we continue to find that variation in risk aversion (as proxied by the measure of Bekaert et al., 2019), and macroeconomic uncertainty are important drivers of the economic gains.

The results in Appendix Table D.3 show the case for yield-only based forecasts. The results substantiate further our conclusion that the relation between utility gains from our portfolio analysis is the strongest with risk aversion and time-varying uncertainty.

In all, the evidence in Table 7 for our forecasts of excess bond returns, and Table D.2 and D.3 for the risk-adjusted economic gains, paints a consistent picture of a significant link between time-varying risk aversion and risk, and bond risk premia. This link is obfuscated when using future realized returns but becomes apparent when using expected returns obtained from machine learning methods. Therefore, models like Bekaert et al. (2009) and Creal and Wu (2018) that combine time variation in economic uncertainty with changes in risk aversion seem to be a promising avenue for research.

[4] Gargano et al. (2019) call the cross-sectional inter-quartile range in GDP and CPI forecasts uncertainty. We follow Buraschi et al. (2019) and refer to the cross-sectional inter-quartile range in forecasts as to disagreement.

TABLE D.2: **Drivers of Portfolio Performance (Forecasts based on Forward Rates + Macro Variables)**

This table reports the regression estimates of economic gain obtained from a mean-variance (panel A) and power utility (panel B) on a set of structural determinants of risk premia (see discussion in the paper for details). The economic gains are based on the predictions implied by the best performing neural network when both yields and macroeconomic variables are considered as predictors, namely the *NN 1 Layer Group Ensem + fwd rate net* (see Table 2 for reference). We standardize both left and right hand variables, so that a 1-standard deviation change in the right hand variables implies a β-standard deviation in the dependent variable. We report the regression estimates as well as Newey-West p-values. Bold font indicates significance at the 5% level. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Panel A: Mean-variance utility

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	0.07	0.05							6.76
	(0.06)	(0.14)							
(ii)			0.03						15.51
			(0.65)						
(iii)				**0.02**					6.59
				(0.01)					
(iv)					**0.07**	**0.08**			25.61
					(0.00)	(0.00)			
(v)							0.06	-0.01	0.50
							(0.64)	(0.75)	
(vi)		0.03	0.03	**0.02**	**0.08**	**0.09**			29.52
		(0.25)	(0.78)	(0.01)	(0.00)	(0.00)			

Panel B: Power utility

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	-0.01	**0.03**							10.27
	(0.34)	(0.02)							
(ii)			**0.56**						20.84
			(0.00)						
(iii)				**0.02**					9.18
				(0.00)					
(iv)					**0.08**	**0.13**			10.16
					(0.04)	(0.00)			
(v)							0.02	-0.00	0.90
							(0.14)	(0.99)	
(vi)		0.02	0.04	**0.02**	**0.03**	**0.02**			33.77
		(0.29)	(0.06)	(0.00)	(0.01)	(0.01)			

TABLE D.3: **Drivers of Portfolio Performance (Forecasts based on Forward Rates only)**

This table reports the regression estimates of economic gain obtained from a mean-variance (panel A) and power utility investor (panel B) on a set of structural determinants of risk premia (see discussion in the paper for details). The economic gains are based on the predictions implied by the best performing neural network when only yields are considered as predictors, namely the *NN 1 Layer (3 nodes)* (see Table 1 for reference). We standardize both left and right hand variables, so that a 1-standard deviation change in the right hand variables implies a β-standard deviation in the dependent variable. We report the regression estimates as well as Newey-West p-values. Bold font indicates significance at the 5% level. The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Panel A: Mean-variance utility

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	**0.06**	0.03							12.30
	(0.03)	(0.12)							
(ii)			**0.02**						10.41
			(0.00)						
(iii)				**0.01**					6.99
				(0.01)					
(iv)					**0.04**	**0.12**			8.28
					(0.00)	(0.00)			
(v)							0.01	-0.01	-1.16
							(0.72)	(0.38)	
(vi)		0.03	0.01	**0.02**	**0.03**	**0.03**			18.12
		(0.14)	(0.07)	(0.02)	(0.00)	(0.01)			

Panel B: Power utility

	$DiB(g)$	$DiB(\pi)$	$-Surplus$	$RAbex$	$UnC(g)$	$UnC(\pi)$	$TYVIX$	$\sigma_B^{(n)}$	$R^2(\%)$
(i)	-0.02	0.02							15.58
	(0.12)	(0.09)							
(ii)			**0.38**						20.38
			(0.00)						
(iii)				**0.01**					9.18
				(0.03)					
(iv)					**0.09**	**0.10**			7.24
					(0.02)	(0.02)			
(v)							**0.02**	-0.01	6.47
							(0.02)	(0.12)	
(vi)		0.02	0.05	**0.01**	**0.03**	**0.02**			30.08
		(0.09)	(0.09)	(0.01)	(0.01)	(0.01)			

E Algorithmic Procedures

In this section we provide details on the algorithmic procedures used for each class of models implemented in the main empirical analysis. We start from the simple penalized regressions, e.g., lasso, ridge, and elastic net. We then turn to non-linear methods starting with shallow regression trees and random forest. We conclude by discussing the different neural network specifics.

E.1 Partial Least Squares

Following the extant practice (see, Ch.3.5 Friedman et al., 2001) Partial Least Squares (PLS) is constructed iteratively as a two-step procedure: in the first step we regress excess bond returns on each predictor $j = 1, \ldots, p$ separately and store the regression coefficient ψ_j. The first partial least squares direction is constructed by multiplying the vector of coefficients by the original inputs, that is $\boldsymbol{x}_1 = \boldsymbol{\psi}' \boldsymbol{y}_t$. Hence the construction of $\boldsymbol{x}_1$ is weighted by the strength of the relationship between the excess bond returns and the predictors. In the second step, excess bond returns are regressed onto $\boldsymbol{x}_1$ giving the coefficient θ_1. Then all inputs are orthogonalized with respect to $\boldsymbol{x}_1$. In this manner, PLS produces a sequence of $l < p$ derived inputs (or directions) orthogonal to each other.[5]

Notice that since the response variable is used to extract features of the input data, the solution path of PLS represents a non-linear function of excess bond returns. Stone and Brooks (1990) and Frank and Friedman (1993) show that, unlike PCA which seeks directions that maximize only the variance, the PLS maximizes both variance and correlation with the response variable subject to orthogonality conditions across derived components.[6] PLS does not require the calibration of hyper-parameters as the derived input directions are deterministically obtained by the two-step procedure outlined above. In this respect, unlike penalized regressions no shrinkage/regularization parameters are required to be calibrated.

E.2 Penalized Regressions

We present the algorithms utilized to estimate the penalized regression models, i.e. the ridge, lasso and elastic net regressions. To recall, penalized regressions add a penalty term $\phi(\boldsymbol{\beta}; \cdot)$ to the least squares loss function $\mathcal{L}(\boldsymbol{\theta})$ where $\boldsymbol{\theta} = (\alpha, \boldsymbol{\beta}^\top)$. The penalty terms in the individual methods are given by

$$\phi(\boldsymbol{\beta}; \cdot) = \begin{cases} \lambda \sum_{j=1}^{p} \beta_j^2 & \text{Ridge regression} \quad \text{(E.1a)} \\ \lambda \sum_{j=1}^{p} |\beta_j| & \text{Lasso} \quad \text{(E.1b)} \\ \lambda\mu \sum_{j=1}^{p} \beta_j^2 + \dfrac{\lambda(1-\mu)}{2} \sum_{j=1}^{p} |\beta_j| & \text{Elastic net} \quad \text{(E.1c)} \end{cases}$$

[5] It is easy to see that for $l = p$ we go back to usual linear least squares estimates similar to PCR.

[6] In particular, the m-th direction solves:

$$\max_{\gamma} \ \text{Corr}^2\left(xr^{(n)}, \boldsymbol{y}\gamma\right) \cdot \text{Var}\left(\boldsymbol{y}\gamma\right)$$
$$\text{subject to} \quad \|\gamma\| = 1, \quad \gamma' \Sigma \hat{\psi}_j = 0, \quad j = 1, \ldots, m-1$$

Apart from the estimation of $\boldsymbol{\theta}$, we have to determine the level of the shrinkage / regularization parameters λ and μ. Usually, this is achieved by cross-validation, i.e. λ and μ are chosen from a suitably wide range of values by evaluating the pseudo out-of-sample performance of the model on a validation sample and picking the λ, μ that yield the best validation error. In the context of time series forecasts the validation sample should be chosen to respect the time-dependence of the observed data, meaning that the validation sample is chosen to follow upon the training sample used to obtain $\boldsymbol{\theta}$ in time. In the following we discuss in more detail the algorithms that are used to obtain coefficient estimates for the penalized regression models.

In contrast to ridge, which is discussed further below, lasso and elastic net coefficient estimates cannot be obtained analytically because of the L^1 component that enters their respective penalty terms (c.f. Eq. (E.1b) and (E.1c)). Hence, we estimate $\boldsymbol{\theta}$ by means of cyclical coordinate descent proposed by Wu et al. (2008) and extended in Friedman et al. (2010). In our exposition of the algorithm of Friedman et al. (2010) we focus on the elastic net case since the lasso case is contained as a special case therein (i.e. by setting $\mu = 0$).

At a high-level coordinate descent can be described as an optimization method aimed at minimizing a loss function one parameter at a time while keeping all other parameters fixed. More precisely, consider the loss function for the elastic net

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - \alpha - \boldsymbol{x}_i^\top \boldsymbol{\beta})^2 + \lambda\mu \sum_{j=1}^{p} \beta_j^2 + \frac{\lambda(1-\mu)}{2} \sum_{j=1}^{p} |\beta_j| \tag{E.2}$$

where the factor two in the denominator in front of the sum is introduced to simplify subsequent expressions for the gradient of the loss function. The minimization of the loss function is unaffected by multiplication with a scalar. Denote by $\mathcal{L}(\boldsymbol{\theta})^{(k)}$ the loss function after the k-th optimization step. The gradient of the loss function with respect to β_j evaluated at its current estimate $\hat{\beta}_j^{(k)}$ is given by

$$-\frac{1}{N} \sum_{i=1}^{N} x_{ij} (y_i - \alpha - \boldsymbol{x}_i^\top \hat{\beta}) + \lambda(1-\mu)\beta_j + \lambda\mu \tag{E.3}$$

if $\hat{\beta}_j > 0$. A similar expression can be obtained for the case $\hat{\beta}_j < 0$ and $\hat{\beta}_j = 0$ (c.f. Friedman et al., 2007). Then, it can be shown that the optimal $\boldsymbol{\beta}$ is obtained by following the Algorithm (1). Commonly, a "warm-start" approach is used to obtain the parameter estimates over the range for λ and μ during cross-validation, meaning that when moving from one set of regularization parameters λ, μ to the next, the prior estimates $\hat{\boldsymbol{\beta}}$ are utilized as initial parameters for the subsequent coordinate descent optimization.

Algorithm 1: Coordinate Descent

Choose initial estimates for $\hat{\alpha} = \bar{y}$ and $\hat{\boldsymbol{\beta}}^{(0)}$ for given λ and μ, where $\bar{y}$ is the unconditional mean of y.
Standardize the inputs x_{ij} such that $\sum_{i=1}^{N} x_{ij} = 0, \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2$, for $j = 1, \ldots, p$.
Set ϵ to desired convergence threshold
while *there is an improvement in the loss function, i.e.* $|\mathcal{L}(\boldsymbol{\theta})^{(k+1)} - \mathcal{L}(\boldsymbol{\theta})^{(k)}| > \epsilon$ **do**
  for *all predictors* $j = 1, \ldots, p$ **do**
    $\hat{y}_i^{(j)} = \hat{\alpha} + \sum_{l \neq j} x_{il}\hat{\beta}_l$, i.e. the fitted value when omitting the covariate x_{ij}
    $\hat{\beta}_j \leftarrow \frac{S\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij}(y_i - \hat{y}_i^{(j)}), \lambda\mu\right)}{1+(1-\mu)}$, defines the parameter-wise update, where S, the soft-thresholding operator, is given by $S(a,b) = \begin{cases} a-b, \text{ if } a > 0 \vee b < |a| \\ a+b, \text{ if } a < 0 \vee b < |a| \\ 0, b \geq a \end{cases}$
  end
end
Output: Estimates $\hat{\beta}$ for given level of λ, μ;

In contrast to lasso and elastic net regressions, the Ridge regression has a closed-form solution given by (e.g., see Friedman et al., 2001, Ch. 3)

$$\hat{\beta}^{\text{Ridge}} = (\boldsymbol{X}^\top \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^\top \boldsymbol{y} \tag{E.4}$$

where $\boldsymbol{X}$ is the input $N \times p$ matrix of p regressors, $\boldsymbol{I}$ is an $N \times N$ identity matrix and $\boldsymbol{y}$ is the vector of dependent variables.

Although, there exists an elegant analytical solution to the ridge regression setup, it is common to apply a matrix decomposition technique to circumvent issues incurred by matrix inversion. Thus, we use a singular value decomposition (SVD) of the matrix $\boldsymbol{X}$ with the form

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top \tag{E.5}$$

where $\boldsymbol{U}$ is an $N \times N$ orthogonal matrix, $\boldsymbol{V}$ is an $p \times p$ orthogonal matrix and D is an $N \times p$ diagonal matrix containing the singular values of $\boldsymbol{X}$. Then it can be shown that the fitted values are given as

$$\boldsymbol{X}\hat{\beta}^{\text{Ridge}} = \boldsymbol{U}\boldsymbol{D}\left(\boldsymbol{D}^2 + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{D}\boldsymbol{D}^\top \boldsymbol{y}. \tag{E.6}$$

The shrinkage parameter λ is chosen by cross-validation. Alternative estimation approaches such as conjugate gradient descent (Zou and Hastie, 2005) become relevant when $\boldsymbol{X}$ gets larger in dimension.

E.3 Tree-Based Methods

Regression trees can approximate any a priori unknown function while keeping the interpretation from a recursive binary tree. However, with more than two inputs, the interpretation is less obvious as trees like the one depicted in Figure 1 grow exponentially in size. Nevertheless the algorithmic procedure is equivalent. Suppose one deals with a partition of M regions $\mathcal{A} = \{A_1, \ldots, A_M\}$ of the

vector of yields $\boldsymbol{y}_t$ such that

$$g(\boldsymbol{y}_t; N) = \sum_{m=1}^{M} \beta_m \mathbb{I}(\boldsymbol{y}_t \in A_m) \ .$$

By minimizing the sum of squared residuals, one can show that the optimal estimate $\hat{\beta}_m$ is just the average of the excess bond returns in that region, i.e., $\hat{\beta}_m = E\left[xr_{t+1}^{(1)} \middle| \boldsymbol{y}_{1:t} \in A_m\right]$. Finding the optimal partition by using a least squares procedure is generally infeasible, however. We thus follow Friedman (2001) and implement a gradient boosting procedure. Gradient boosting in a tree context boils down to combining several weak trees of shallow depth.

Boosting is a technique for reducing the variance of the model estimates and increasing precision. However, trees are "grown" in an adaptive way to reduce the bias, and thus are not identically distributed. An alternative procedure would be to build a set of *de-correlated* trees which are estimated separately and then averaged out. Such modeling framework is known in the machine learning literature as "Random Forests" (see Breiman, 2001). It is a substantial modification of bagging (or bootstrap aggregation) whereby the outcome of independently drawn processes is averaged to reduce the variance estimates. Bagging implies that the regression trees are identically distributed – that is the variance of the average estimates, as the number of simulated trees increases, depends on the variance of each tree times the correlation among the trees. Random forests aim to minimize the variance of the average estimate by minimizing the correlation among the simulated regression trees.

We also consider an extended version of the random forest procedure which is called "Extremely Randomized Trees" (Geurts et al., 2006) . While similar to ordinary random forests, in that they still represent an ensemble of individual trees, extreme trees have two main distinguishing features: first, each tree is trained using the whole training sample (rather than a bootstrap sample); and second, the top-down splitting in the tree learner is randomized. That means that instead of computing the optimal cut-point locally for each input variable under consideration, a random cut-point is selected. In other words, with extreme trees the split of the trees is stochastic; with random forests the split is instead deterministic.

Tree-based methods such as Gradient Boosted Regression Trees or Random Forests are essentially modifications of a universal underlying algorithm utilized for the estimation of regression trees, commonly, that is the Classification and Regression Tree (CART) algorithm (Breiman et al., 1984) presented in Algorithm (2).

Algorithm 2: Classification and Regression Trees

Initialize tree $T(D)$ where D denotes the depth; denote by $R_l(d)$ the covariates in branch l at depth d.

for $d = 1, \dots, D$ **do**

  for $\tilde{R}$ in $\{R_l(d), l = 1, \dots, 2^{d-1}\}$ **do**

    Given splitting variable j and split point s define regions

$$R_{\text{left}}(j,s) = \{X \mid X_j \leq s, X_j \cap \tilde{R}\} \quad \text{and} \quad R_{\text{right}}(j,s) = \{X \mid X_j > s, X_j \cap \tilde{R}\}$$

    In the splitting regions set

$$c_{\text{left}}(j,s) \leftarrow \frac{1}{|R_{\text{left}(j,s)}|} \sum_{x_i \in R_{\text{left}(j,s)}} y_i(x_i) \quad \text{and} \quad c_{\text{right}}(j,s) \leftarrow \frac{1}{|R_{\text{right}(j,s)}|} \sum_{x_i \in R_{\text{right}(j,s)}} y_i(x_i)$$

    Find j^*, s^* that optimize

$$j^*, s^* = \underset{j,s}{\text{argmin}}\Big[\sum_{x_i \in R_{\text{left}(j,s)}} (y_i - c_{\text{left}}(j,s))^2 + \sum_{x_i \in R_{\text{right}(j,s)}} (y_i - c_{\text{right}}(j,s))^2 \Big]$$

    Set the new partitions

$$R_{2l}(d) \leftarrow R_{\text{right}}(j^*, s^*) \quad \text{and} \quad R_{2l-1}(d) \leftarrow R_{\text{left}}(j^*, s^*)$$

  end

end

Output: A fully grown regression tree T of depth D. The output is given by

$$f(x_i) = \sum_{k=1}^{2^L} \text{avg}(y_i \mid x_i \in R_k(D)) \mathbf{1}_{\{x \in R_k(D)\}},$$

i.e. the average response in each region R_k at depth D.

Next, we present the Algorithm (3) used to populate random forests as suggested by Breiman (2001). Random Forests consist of trees populated following an algorithm like CART, but randomly select a sub-set of predictors from the original data. In this manner, the individual trees in the forest are de-correlated and overall predictive performance relative to a single tree is increased. The hyperparameters to be determined by cross-validation include first and foremost the number of trees in the forest, the depth of the individual trees and the size of the randomly selected sub-set of predictors. Generally, larger forests tend to produce better forecasts in terms of predictive accuracy. Finally, Algorithm (4) delivers the gradient boosted regression tree (GBRT) (Friedman, 2001). GBRTs are based on the idea of combining the forecasts of several weak learners. The GBRT comprises of trees of shallow depth that produce weak predictions stand-alone, however, tend to deliver powerful forecasts when aggregated adequately.

Algorithm 3: Random Forest

Determine forest size F

for $t = 1, \ldots, F$ **do**

Obtain bootstrap sample Z from original data.

Grow full trees following Algorithm (2) with the following adjustments:

1. Select $\tilde{p}$ variables from the original set of p variables.
2. Choose the best combination (j, s) (c.f. Algorithm (2)) from $\tilde{p}$ variables
3. Create the two daughter nodes

Denote the obtained tree by T_t

end

Output: Ensemble of F many trees. The output is the average over the trees in the forest given as

$$f(x_i) = \frac{1}{F} \sum_{t=1}^{F} T_t(x_i)$$

Algorithm 4: Gradient Boosted Regression Trees

Initialize a gradient boosted regression tree $f_0(x) = 0$ and determine number of learners F. Let $\mathcal{L}(y, f(x))$ be the loss function associated with tree output $f(x)$.

for $t = 1, \ldots, F$ **do**

for $i = 1, \ldots, N$ **do**

Compute negative gradient of loss function evaluated for current state of regressor $f = f_{t-1}$

$$r_{it} = -\frac{\partial \mathcal{L}(y_i, f_{t-1}(x_i))}{\partial f_{t-1}(x_i)}.$$

end

Using the just obtained gradients grow a tree of depth D (commonly, $D \ll p$ where p is the number of predictors) on the original data replacing the dependent variable with $\{r_{it}, \forall i\}$. Denote the resulting predictor as $g_t(x)$.

Update the learner f_t by

$$f_t(x) \leftarrow f_{t-1}(x) + \nu g_t(x)$$

where $\nu \in (0, 1]$ is a hyperparameter.

end

Output: $f_F(x)$ is the gradient boosted regression tree output.

E.4 Neural Networks

A commonly used algorithm to fit neural networks is stochastic gradient descent (SGD). For this paper we make use of a modified form of gradient decent by adding Nesterov momentum (Nesterov, 1983). In comparison to plain SGD which is often affected by oscillations between local minima, Nesterov momentum (also known as Nesterov accelerated gradient) accelerates SGD in the relevant direction. Algorithm (5) outlines the procedure. It is best practice to initialize neural network parameters with zero mean and unit variance or variations thereof such as He et al. (2015b) like we do in this paper. Over the course of the training process this normalization vanishes and a problem referred to as covariate shift occurs. Thus, we apply batch normalization (Ioffe and Szegedy, 2015) to the activations after the last ReLU layer.

Algorithm 5: Stochastic Gradient Decent with Nesterov Momentum

Initialize the vector of neural network parameters θ_0 and choose momentum parameter γ. Determine the learning rate η and set $v_0 = 0$.

while *No convergence of* θ_t **do**

$t \leftarrow t + 1$

$v_t = \gamma v_{t-1} + \eta \nabla_{\theta_t} \mathcal{L}(\theta_t)$

$\theta_t \leftarrow \theta_{t-1} - v_t$

end

Output: The parameter vector θ_t of the trained network.

Batch normalization reduces the amount of variability of predictors by adjusting and scaling the activations. This increases the stability of the neural network and the speed of training. The output of a previous activation is normalized by subtracting the batch mean and dividing by the batch standard deviation. This is particularly advantageous if layers without activation functions, i.e. the output layer, follow layers with non-linear activations, such as ReLU, which tend to change the distributions of the activations. Since the normalization is applied to each individual mini-batch, the procedure is referred to as batch normalization. The SGD optimization remains largely unaffected; in fact, by using batch normalization the structure of the weights is much more parsimonious. Algorithm (6) outlines the procedure from the original paper.

Algorithm 6: Batch Normalization per mini-batch

Let $\mathcal{B} = x_{1,\dots,m}$ be a mini-batch of batch-size m. Set parameters λ, β.

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$

$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$

$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}}$

$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$

Output: The normalized mini-batch $mathrmBN_{\gamma,\beta}(x_i)$.

Algorithm (7) presents the early stopping procedure that is used to abort the training process early when the loss on the validation sample has not improved for a specific number of consecutive iterations. Early stopping is used to improve the performance of the trained models and reduce over-fitting. By evaluating the validation error it prevents the training procedure from simply memorizing the training data (see Bishop, 1995 and Goodfellow et al., 2016). More specifically, by means of early stopping the training process is stopped prematurely if the loss on the validation sample has not improved for a number of consecutive epochs. In detail, our algorithm is stopped early if any of the following is

true: maximum number of epochs reached the value of 1000, gradient of loss function falls below a specified threshold, or the MSE on validation set has not improved for 20 consecutive epochs. When early stopping occurs we retrieve the model with the best validation performance. Early stopping has two effects. Firstly, early stopping prevents over-fitting by aborting the training when the pseudo out-of-sample performance starts to deteriorate, hence it reduces over-fitting. Secondly, since the optimal number of weight updates is unknown initially, early stopping helps to keep the computational cost at a minimum by potentially stopping the training far before the maximum number of iterations is reached.

Algorithm 7: Early Stopping

Initialize the validation error $\epsilon = \inf$ and define a patience ϕ, also set $k = 0$
while $k < \phi$ **do**
  Update θ using Algorithm (5) to get $\theta^{(j)}$, i.e. the parameter vector at iteration j
  Compute loss function on validation sample $\epsilon' = \mathcal{L}_{val}(\theta; \cdot)$ **if** $\epsilon > \epsilon$ **then**
    $j \leftarrow j + 1$
  end
  else
    $j \leftarrow 0$
    $\epsilon \leftarrow \epsilon'$
    $\theta' = \theta^{(j-p)}$
  end
end
Output: The early-stopping optimized parameter vector θ'

Finally, it is important to highlight that we use a form of forecast averaging / ensembling, i.e. we train multiple copies of networks with different seeds for the randomly drawn initial network weights. Using fixed seeds will in general lead to replicable results. Nevertheless, different seeds will produce different forecasts as discussed also in Gu et al. (2018). Therefore, in order to reduce prediction variance, we average over forecasts from networks initialized with different seeds. To be precise, for each time t we initialize 100 models with different but fixed seeds. The 100 models are then trained and as part of the training we obtain the validation sample loss. The validation sample loss is then used to select the 10 out 100 models with the smallest validation sample error. Finally, we average the forecasts of those 10 in-sample best performing models.

F Computational Details

For our implementation of the various machine learning techniques in Python we utilize the well-known packages `Scikit-Learn`[7] and `Tensorflow`[8] in the `Keras`[9] wrapper. `Scikit-Learn` provides the functionality to estimate regression trees (both gradient boosted regression trees and random forest), partial least squares and penalized regressions (ridge, lasso, elastic net). Furthermore, we make use of numerous auxiliary functions from `Scikit-Learn` for data pre-processing such as input standardization / scaling and train-test splits. A particularly useful `Scikit-Learn` function is `GridSearchCV`, which allows streamlined systematic investigation of neural network hyperparameters. Our neural networks are trained using `Keras` and Google's `Tensorflow`. The `Keras` wrapper provides

[7] `http://scikit-learn.org/stable/`, as of 26th October 2018
[8] `https://www.tensorflow.org/`, as of 26th October 2018
[9] `https://keras.io/`, as of 26th October 2018

two distinct approaches to construct neural networks, i.e. a sequential API and a functional API. The sequential API is sufficient to construct relatively simple network structures that do not require merged layers, while the functional API is used to build those networks that require merged layers as for example in the case of the exogenous addition of forward rates into the last hidden layer. `Keras` also implements a wide range of regularization methods applied in this paper, i.e. early-stopping by cross-validation, L1 / L2 penalties, drop-out, and batch normalization.

F.1 Setup

Since the forecasting exercise in this paper is iterative and since we use model averaging, the computational challenge becomes sizable. For that reason, we perform all computations on a high performance computing cluster consisting of 84 nodes with 28 cores each, totaling to more than 2300 cores. We parallelize our code using the Python `multiprocessing`[10] package. Specifically, we parallelize our code execution at the point of model averaging such that for each forecasting step a large number of models can be estimated in parallel and averaged before moving to the next time step. Although it is common in applications such as image recognition to perform neural network training on GPUs, we refrain from doing so since the speed-up from GPU computing would be eradicated by the increased communication over-head between CPU and GPU as the computational effort of training an individual neural network is relatively small in our exercise.

F.2 Full Cross-Validated Neural Network vs. Group-Ensemble

In Section C.5 we compare our best performing group-ensemble model (labeled as NN 1 Layer Group Ensem (1 node per group), fwd rate net (1 layer: 3 nodes)) against two different types of model averaging schemes, i.e., weighting based on the inverse of the validation loss and an equal-weighted model, as well as a fully cross-validated (CV) network. While the logic of the weighting schemes for the model-averaging may be intuitive, the specifications for the fully CV network may be not. Table F.1 gives an outline of these specifications vis-a-vis the choice made for the group-ensemble structure.

A number of aspects should be discussed; in the full CV setting we recursively choose the number of hidden layers (between 1 and 2) as well as the number of nodes for each group of macroeconomic variables. Similarly, the number of nodes in the forward rate net is allowed to change whereas for the group-ensemble it is kept fixed. In addition, the full CV allows for a bit more flexibility in the L1/L2 penalty terms for both the group-specific networks as well as for the forward rates. Increasing the flexibility comes at the cost of increased computational expense (423 possible combinations vs. 6). To keep the computational cost manageable we employ a procedure referred to as randomized cross-validation that randomly draws combinations of hyperparameters from the set of available combinations. Specifically, we perform randomized cross-validation over 60 specifications from the the set of hyperparameters above. Note, even under randomized cross-validation as we employ it here the effective training time increases by a factor of 10 (60 specifications vs 6 specifications).

[10] `https://docs.python.org/3.4/library/multiprocessing.html`, as of 26th October 2018

TABLE F.1: **Specifics of Full Cross-Validation vs. Group-Ensembling**

This table reports the hyperparameters for the full cross-validated network vs. the shallow network with group ensemble (see Table C.3). The out-of-sample predictions are obtained by a recursive forecast which starts in January 1990. The sample period is from 1971:08-2018:12.

Hyperparameter Set	Full CV	CV for group ensemble
Number of Layers	1, 2	1
Nodes Per Group	1, 2, 3	1
Nodes in Fwd Rate Net	1, 3	3
Dropout - Group Net	0.1, 0.3, 0.5	0.1, 0.3, 0.5
Dropout - Fwd Rate Net	0, 0.3	0
L1/L2 Penal - Group Net	0.01, 0.001, 0.0001	0.01, 0.001
L1/L2 Penal - Fwd Rate Net	0.001, 0.0001	0.0001
Combinations per retraining	432	6
CV frequency	5 years	5 years